

Guia Foca GNU/Linux

Capítulo 5 - Discos e Partições

Este capítulo traz explicações de como manipular discos rígidos e partições no sistema GNU/Linux e como acessar seus discos de CD-ROM e partições DOS, Windows 9X/XP/Vista no GNU/Linux.

Também será ensinado como formatar uma partição ou arquivo em formato EXT2, EXT3, reiserfs e usar a ferramenta `mkswap` (para criar uma partição ou arquivo de memória virtual).

5.1 Partições

São divisões existentes no disco rígido que marcam onde começa onde termina um sistema de arquivos. Por causa destas divisões, nós podemos usar mais de um sistema operacional no mesmo computador (como o GNU/Linux, Windows e DOS), ou dividir o disco rígido em uma ou mais partes para ser usado por um único sistema operacional.

Para gravar os dados, o disco rígido deve ser primeiro particionado (usando o `fdisk`), escolher o tipo da partição (*Linux Native*, *Linux Swap*, etc) e depois aquela partição deve ser formatada com o `mkfs.ext2` (veja [Partição EXT2 \(Linux Native\), Seção 5.3](#)).

Após criada e formatada, a partição será identificada como um dispositivo no diretório `/dev` (veja [Identificação de discos e partições em sistemas Linux, Seção 5.12](#)) . e deverá ser montada ([Montando \(acessando\) uma partição de disco, Seção 5.13](#)) para permitir seu uso no sistema.

Uma partição de disco não interfere em outras partições existentes, por este motivo é possível usar o Windows, GNU/Linux e qualquer outro sistema operacional no mesmo disco. Para escolher qual deles será inicializado, veja [Gerenciadores de Partida \(boot loaders\), Capítulo 6](#).

Para particionar (dividir) o disco rígido em uma ou mais partes é necessário o uso de um programa de particionamento. Os programas mais conhecidos para particionamento de discos no GNU/Linux são `fdisk`, `cfdisk` e o `Disk Druid`.

Lembre-se:

- Quando se apaga uma partição, você estará apagando TODOS os arquivos existentes nela!
- A partição do tipo *Linux Native* (Tipo 83) é a usada para armazenar arquivos no GNU/Linux. Para detalhes veja [Partição EXT2 \(Linux Native\), Seção 5.3](#).
- A partição do tipo *Linux Swap* (Tipo 82) é usada como memória virtual. Para detalhes veja [Partição Linux Swap \(Memória Virtual\), Seção 5.7](#).
- Em sistemas novos, é comum encontrar o Windows instalado em uma partição que consome TODO o espaço do disco rígido. Uma solução para instalar o GNU/Linux é apagar a partição Windows e criar três com tamanhos menores (uma para o Windows, uma para o GNU/Linux e outra para a *Memória Virtual do Linux* (SWAP). Ou criar apenas 2 se você não quiser mais saber mais do Windows ;-)

A outra é usar o `parted` (e `gparted` sua versão gráfica), um bom exemplo de programa de reparticionamento não destrutivo (que usa o espaço livre existente) que trabalha com `FAT16`, `FAT32`, `NTFS`. Esta técnica também é chamada de Reparticionamento não destrutivo (e o outro obviamente Reparticionamento destrutivo).

Para mais detalhes sobre discos, partições ou como particionar seu disco, veja algum bom documento sobre particionamento (como a página de manual e documentação do `fdisk`, `cfdisk`, `parted` ou `Disk Druid`).

5.2 Sistema de Arquivos

É criado durante a "formatação" da partição de disco (quando se usa o comando `mkfs.ext2`). Após a formatação toda a estrutura para leitura/gravação de arquivos e diretórios pelo sistema operacional estará pronta para ser usada. Normalmente este passo é feito durante a instalação de sua distribuição `GNU/Linux`.

Cada sistema de arquivos tem uma característica em particular mas seu propósito é o mesmo: Oferecer ao sistema operacional a estrutura necessária para ler/gravar os arquivos/diretórios.

Entre os sistemas de arquivos existentes posso citar:

- `Ext2` - Usado em partições *Linux Nativas* para o armazenamento de arquivos. É identificado pelo código 83. Seu tamanho deve ser o suficiente para acomodar todos os arquivos e programas que deseja instalar no `GNU/Linux` (você encontra isto no manual de sua distribuição). Para detalhes veja [Partição EXT2 \(Linux Native\), Seção 5.3](#).
 - `Ext3` - Este sistema de arquivos possui melhorias em relação ao `ext2`, como destaque o recurso de `journaling`. Ele também é identificado pelo tipo 83 e totalmente compatível com o `ext2` em estrutura. O `journal` mantém um log de todas as operações no sistema de arquivos, caso aconteça uma queda de energia elétrica (ou qualquer outra anormalidade que interrompa o funcionamento do sistema), o `fsck` verifica o sistema de arquivos no ponto em que estava quando houve a interrupção, evitando a demora para checar todo um sistema de arquivos (que pode levar minutos em sistemas de arquivos muito grandes). Para detalhes veja [Partição EXT3 \(Linux Native\), Seção 5.5](#).
 - `Swap` - Usado em partições *Linux Swap* para oferecer memória virtual ao sistema. Note que é altamente recomendado o uso de uma partição `Swap` no sistema (principalmente se você tiver menos que 16MB de memória RAM). Este tipo de partição é identificado pelo código 82. Para detalhes veja [Partição Linux Swap \(Memória Virtual\), Seção 5.7](#).
 - `proc` - Sistema de arquivos do kernel (veja [O sistema de arquivos /proc, Seção 5.8](#)).
 - `FAT12` - Usado em disquetes no DOS
 - `FAT16` - Usado no DOS e oferece suporte até discos de 2GB
 - `FAT32` - Também usado no DOS e oferece suporte a discos de até 2 Terabytes
-

5.3 Partição EXT2 (Linux Native)

A partição `EXT2` é o tipo usado para criar o sistema de arquivos `Linux Native` usado para armazenar o sistema de arquivos `EXT2` (após a formatação) e permitir o armazenamento de dados. Para detalhes de como criar uma partição `EXT2` veja [Criando um sistema de arquivos EXT2 em uma partição, Seção 5.3.1](#).

Este tipo de partição é normalmente identificado pelo código 83 nos programas de particionamento de disco. Note que também é possível criar um sistema de arquivos `EXT2` em um arquivo (ao invés de uma partição) que poderá ser montado e acessado normalmente pelo sistema de arquivos (veja [Criando um sistema de arquivos EXT2 em um arquivo, Seção 5.3.2](#)).

Logo que foi inventado, o `GNU/Linux` utilizava o sistema de arquivos `Minix` (e conseqüentemente uma partição `Minix`) para o armazenamento de arquivos. Com a evolução do desenvolvimento, foi criado o padrão `EXT` (*Extended Filesystem*) e logo evoluiu para o `EXT2` (*Second Extended Filesystem*) que é o usado hoje em dia.

Você deve escolher este tipo de partição para armazenar seus arquivos, é o padrão atualmente, é o mais rápido, não se fragmenta tão facilmente pois permite a localização do melhor lugar onde o arquivo se encaixa no disco, etc. Isto é útil para grandes ambientes multiutilizador onde várias pessoas gravam/apagam arquivos o tempo todo.

5.3.1 Criando um sistema de arquivos EXT2 em uma partição

O utilitário usado para formatar uma partição `EXT2` é o `mkfs.ext2`. Após terminar este passo, seu sistema de arquivos `EXT2` estará pronto para ser usado.

Após particionar seu disco rígido e criar uma (ou várias) partições `EXT2`, use o comando:

```
mkfs.ext2 /dev/hda?
```

Onde a "?" em `hda?` significa o número da partição que será formatada. A identificação da partição é mostrada durante o particionamento do disco, anote se for o caso. `hda` é o primeiro disco rígido IDE, `hdb` é o segundo disco rígido IDE. Discos SCSI são identificados por `sda?`, `sdb?`, etc. Para detalhes sobre a identificação de discos, veja [Identificação de discos e partições em sistemas Linux, Seção 5.12](#).

Algumas opções são úteis ao `mkfs.ext2`:

- `-c` Procura blocos danificados na partição antes de criar o sistema de arquivos.
- `-L NOME` Coloca um nome (label) no sistema de arquivos.
- `-b NUM` Define o tamanho do bloco, em bytes.
- `-m NUM` Define a porcentagem de espaço em disco reservada para manutenção (por padrão reservado para o root, mas isto é alterável).

Agora para acessar a partição deverá ser usado o comando: `mount /dev/hda? /mnt -t ext2`

Para mais detalhes veja [Montando \(acessando\) uma partição de disco, Seção 5.13](#).

Note que é possível criar um sistema de arquivos no disco rígido sem criar uma partição usando `/dev/hda`, `/dev/hdb`, etc. **EVITE FAZER ISSO!** Como não estará criando uma partição, o disco estará dividido de maneira incorreta, você não poderá apagar o sistema de arquivos completamente do disco caso precise (lembre-se que você não criou uma partição), e a partição possui uma assinatura apropriada que identifica o sistema de arquivos.

O espaço padrão reservado na partição para o utilizador root é de 5%. Em sistemas com partições maiores que 3Gb, isso pode representar uma grande quantidade de espaço em disco não utilizada por outros utilizadores.

Veja a opção `-m` sobre como fazer esta modificação. Caso já tenha criado a partição, isto pode ser feito no `tune2fs` com a opção `-m`.

5.3.2 Criando um sistema de arquivos EXT2 em um arquivo

É possível criar um sistema de arquivos EXT2 em um arquivo que poderá ser montado e acessado normalmente como se fosse uma partição normal. Isto é possível por causa do recurso `loop` oferecido pelo kernel do GNU/Linux. Os dispositivos de `loop` estão disponíveis no diretório `/dev` com o nome `loop?` (normalmente estão disponíveis 8 dispositivos de `loop`).

Isto é possível usando o comando `dd` e o `mkfs.ext2`. Veja passo a passo como criar o sistema de arquivos EXT2 em um arquivo:

1. Use o comando `dd if=/dev/zero of=/tmp/arquivo-ext2 bs=1024 count=10000` para criar um arquivo `arquivo-ext2` vazio de 10Mb de tamanho em `/tmp`. Você pode modificar os parâmetros de `of` para escolher onde o arquivo será criado, o tamanho do arquivo poderá ser modificado através de `count`.
2. Formate o arquivo com `mkfs.ext2 /tmp/arquivo-ext2`. Ele primeiro dirá que o arquivo `arquivo-ext2` não é um dispositivo de bloco especial (uma partição de disco) e perguntará se deve continuar, responda com `y`.

O sistema de arquivos EXT2 será criado em `/tmp/arquivo-ext2` e estará pronto para ser usado.

3. Monte o arquivo `arquivo-ext2` com o comando: `mount /tmp/arquivo-ext2 /mnt -o loop=/dev/loop1`. Note que foi usado o parâmetro `-o loop` para dizer ao comando `mount` para usar os recursos de `loop` do kernel para montar o sistema de arquivos.
4. Confira se o sistema de arquivos EXT2 em `arquivo-ext2` foi realmente montado no sistema de arquivos digitando `df -T`. Para detalhes, veja [df, Seção 10.3](#).

Pronto! o que você gravar para `/mnt` será gravado dentro do arquivo `/tmp/arquivo-ext2`. Como foi criado um sistema de arquivos EXT2 em `arquivo-ext2`, você poderá usar todos os recursos da partição EXT2 normal, como permissões de arquivos e diretórios, links simbólicos, etc.

O uso da opção `loop=/dev/loop1` permite que o dispositivo `/dev/loop1` seja associado ao arquivo `/arquivo-ext2` e assim permitir sua montagem e uso no sistema.

- Você poderá usar apenas `-o loop` com o comando `mount`, assim o kernel gerenciará automaticamente os dispositivos de `loop`.
 - Caso faça isto manualmente, lembre-se de usar dispositivos `/dev/loop?` diferentes para cada arquivo que montar no sistema. Pois cada um faz referência a um único arquivo.
-

5.4 Journaling

O sistema de journaling grava qualquer operação que será feita no disco em uma área especial chamada "journal", assim se acontecer algum problema durante a operação de disco, ele pode voltar ao estado anterior do arquivo, ou finalizar a operação.

Desta forma, o journal acrescenta ao sistema de arquivos o suporte a alta disponibilidade e maior tolerância a falhas. Após uma falha de energia, por exemplo, o journal é analisado durante a montagem do sistema de

arquivos e todas as operações que estavam sendo feitas no disco são verificadas. Dependendo do estado da operação, elas podem ser desfeitas ou finalizadas. O retorno do servidor é praticamente imediato (sem precisar a enorme espera da execução do fsck em partições maiores que 10Gb), garantindo o rápido retorno dos serviços da máquina.

Outra situação que pode ser evitada é com inconsistências no sistema de arquivos do servidor após a situação acima, fazendo o servidor ficar em estado 'single user' e esperando pela intervenção do administrador. Este capítulo do guia explica a utilização de journaling usando o sistema de arquivos `ext3` (veja [Partição EXT3 \(Linux Native\), Seção 5.5](#) para detalhes).

5.5 Partição EXT3 (Linux Native)

O sistema de arquivos `ext3` faz parte da nova geração extended file system do Linux, sendo que seu maior benefício é o suporte a journaling.

O uso deste sistema de arquivos comparado ao `ext2`, na maioria dos casos, melhora o desempenho do sistema de arquivos através da gravação seqüencial dos dados na área de metadados e acesso mhash a sua árvore de diretórios.

A estrutura da partição `ext3` é semelhante a `ext2`, o journaling é feito em um arquivo chamado `.journal` que fica oculto pelo código `ext3` na partição (desta forma ele não poderá ser apagado, comprometendo o funcionamento do sistema). A estrutura idêntica da partição `ext3` com a `ext2` torna mais fácil a manutenção do sistema, já que todas as ferramentas para recuperação `ext2` funcionarão sem problemas.

5.5.1 Criando um sistema de arquivos EXT3 em uma partição

Para criar uma partição `ext3`, utilize o comando `mkfs.ext3` ou o `mkfs.ext2` junto com a opção `-j`. As opções usadas pelo `mkfs.ext3` são idênticas a `mkfs.ext2` (documentado em [Criando um sistema de arquivos EXT2 em uma partição, Seção 5.3.1](#)). A única vantagem desta ferramenta comparada ao `mkfs.ext2` é que a opção `-j` é automaticamente adicionada a linha de comando para criar um sistema de arquivos com journal. Se você é daqueles que querem ter um controle maior sobre o tamanho do arquivo de journal, use a opção `-J [tam]` (onde tamanho é o tamanho em Megabytes).

Quando uma partição `ext3` é criada, o arquivo `.journal` é criado no raiz da partição, sendo usado para gravar os metadados das transações de journaling. A estrutura da partição `ext2` não difere em nada da `ext3`, a não ser este arquivo e a opção "has_journal" que é passada a partição.

Por exemplo, para criar uma partição `ext3` em `/dev/hda1`:

```
mkfs.ext3 /dev/hda1
```

ou

```
mkfs.ext2 -j /dev/hda1
```

Basta agora montar a partição com o comando `mount /dev/hda1 /teste -t ext3` (para montar a partição em `/teste`). Após isto, modifique o `/etc/fstab` para montar a partição como `ext3` quando o Linux for

iniciado. Para mais detalhes veja [Montando \(acessando\) uma partição de disco, Seção 5.13.](#)). Caso o suporte a `ext3` tenha sido compilado no kernel, ele tentará detectar e montar a partição como `ext3`, caso contrário, ele usará `ext2`.

Sua partição agora está montada como `ext3`, para conferir digite: `df -T`.

OBS: Quando criar um sistema de arquivos `ext3` em uma partição raiz (/), tenha certeza de incluir o suporte a `ext3` embutido no kernel, caso contrário a partição será montada como `ext2`.

5.5.2 Criando um sistema de arquivos EXT3 em um arquivo

As instruções para criar um sistema de arquivos `ext3` em um arquivo não difere muito das instruções de [Criando um sistema de arquivos EXT2 em um arquivo, Seção 5.3.2](#), apenas utilize a opção `-j` ou `-J` `[tamanho_em_mb]` (como explicado em [Criando um sistema de arquivos EXT3 em uma partição, Seção 5.5.1](#)).

5.5.3 Fazendo a conversão do sistema de arquivos EXT2 para EXT3

Se você já possui um uma partição `ext2` e deseja converte-la para `ext3` isto poderá ser feito facilmente, de forma segura (sem qualquer risco de perda de dados) e você poderá voltar para o sistema `ext2` caso deseje (veja [Convertendo de EXT3 para EXT2, Seção 5.5.4](#)).

Primeiro, execute o comando `tune2fs` na partição que deseja converter com a opção `-j` ou `-J` `[tamanho_journal]` para adicionar o suporte a Journaling na partição. Este comando poderá ser executado com segurança em uma partição `ext2` montada, após converter remonte a partição usando os comandos `umount /particao` e `mount /particao`.

Após a conversão para `ext3` é desnecessária a checagem periódica do sistema de arquivos (que por padrão é após 20 montagens e a cada 30 dias). Você pode desativar a checagem após o número máximo de montagens com a opção `-c` `[num_vezes]`, e o número de dias máximos antes de verificar novamente com a opção `-i` `[num_dias]` (o uso de 0 desativa). Por exemplo:

```
tune2fs -c 0 -i 90 /dev/hda2
```

Desativa a checagem após número máximo de montagens (`-c 0`) e diz para a partição ser verificada a cada 90 dias (`-i 90`).

O último passo é modificar o `/etc/fstab` para que a partição seja montada como `ext3` na inicialização e depois desmontar (`umount /dev/hda2` e remonte-la (`mount /dev/hda2`) para usar o suporte `ext3`. Confira se ela está usando `ext3` usando o comando `df -T`.

OBS: Caso a partição convertida para `ext3` seja a raiz (/), tenha certeza de incluir o suporte a `ext3` embutido no kernel, caso contrário, a partição será montada como `ext2`.

5.5.4 Convertendo de EXT3 para EXT2

Remover o suporte a `ext3` de uma partição é simples, rápido e seguro. Execute os seguintes passos:

- Execute o comando `tune2fs -O^has_journal /dev/hdxx` na partição que deseja remover o Journal. Este comando poderá ser executado em uma partição montada.
- Modifique o `/etc/fstab` e altere a partição para `ext2`.

- Desmonte e monte novamente a partição com os comandos: `umount /dev/hdxx` e `mount /dev/hdxx`.

- Pronto! a partição agora é novamente uma partição *ext2* normal, confira digitando `df -T`.

Pronto, o suporte a *ext3* foi removido do seu sistema e agora poderá usar a partição como *ext2* normalmente (confira digitando `df -T`).

5.6 Sistema de arquivos reiserfs

Este é um sistema de arquivos alternativo ao *ext2/3* que também possui suporte a journaling. Entre suas principais características, estão que ele possui tamanho de blocos variáveis, suporte a arquivos maiores que 2 Gigabytes (esta é uma das limitações do *ext3*) e o acesso mhash a árvore de diretórios é um pouco mais rápida que o *ext3*.

Para utilizar *reiserfs*, tenha certeza que seu kernel possui o suporte habilitado (na seção `File Systems`) e instale o pacote *reiserfsprogs* que contém utilitários para formatar, verificar este tipo de partição.

5.6.1 Criando um sistema de arquivos reiserfs em uma partição

Para criar uma partição *reiserfs*, primeiro instale o pacote *reiserfsprogs* (`apt-get install reiserfsprogs`).

Para criar uma partição *reiserfs*, primeiro crie uma partição *ext2* normal, e então use o comando:

```
mkreiserfs /dev/hda?
```

Onde a "?" em *hda?* significa o número da partição que será formatada com o sistema de arquivos *reiserfs*. A identificação da partição é mostrada durante o particionamento do disco, anote se for o caso. *hda* é o primeiro disco rígido IDE, *hdb* é o segundo disco rígido IDE. Discos SCSI são identificados por *sda?*, *sdb?*, etc. Para detalhes sobre a identificação de discos, veja [Identificação de discos e partições em sistemas Linux, Seção 5.12](#).

Algumas opções são úteis ao *mkreiserfs*:

- `-s [num]` - Especifica o tamanho do arquivo de journal em blocos. O valor mínimo é 513 e o máximo 32749. O valor padrão é 8193.
- `-l [NOME]` - Coloca um nome (label) no sistema de arquivos.
- `-f` - Força a execução do *mkreiserfs*.
- `-d` - Ativa a depuração durante a execução do *mkreiserfs*.

Agora para acessar a partição deverá ser usado o comando: `mount /dev/hda? /mnt -t reiserfs`

Para mais detalhes veja [Montando \(acessando\) uma partição de disco, Seção 5.13](#).

Note que é possível criar um sistema de arquivos no disco rígido sem criar uma partição usando */dev/hda*, */dev/hdb*, etc. usando a opção `-f` **EVITE FAZER ISSO!** Como não estará criando uma partição, o disco estará dividido de maneira incorreta, você não poderá apagar o sistema de arquivos completamente do disco caso precise (lembre-se que você não criou uma partição), e a partição possui uma assinatura apropriada que identifica o sistema de arquivos.

5.6.2 Criando um sistema de arquivos reiserfs em um arquivo

O sistema de arquivos `reiserfs` também poderá ser criado em um arquivo, usando os mesmos benefícios descritos em [Criando um sistema de arquivos EXT2 em um arquivo, Seção 5.3.2](#). Para fazer isso execute os seguintes passos em sequência:

1. Use o comando `dd if=/dev/zero of=/tmp/arquivo-reiserfs bs=1024 count=33000` para criar um arquivo `arquivo-reiserfs` vazio de 33Mb de tamanho em `/tmp`. Você pode modificar os parâmetros de `of` para escolher onde o arquivo será criado, o tamanho do arquivo poderá ser modificado através de `count`. Note que o tamanho mínimo do arquivo deve ser de 32Mb, devido aos requerimentos do `reiserfs`.
2. Formate o arquivo com `mkreiserfs -f /tmp/arquivo-reiserfs`. Ele primeiro dirá que o arquivo `arquivo-reiserfs` não é um dispositivo de bloco especial (uma partição de disco) e perguntará se deve continuar, responda com `y`.

O sistema de arquivos ReiserFS será criado em `/tmp/arquivo-reiserfs` e estará pronto para ser usado.

3. Monte o arquivo `arquivo-reiserfs` com o comando: `mount /tmp/arquivo-reiserfs /mnt -t reiserfs -o loop=/dev/loop1`. Note que foi usado o parâmetro `-o loop` para dizer ao comando `mount` para usar os recursos de `loop` do `kernel` para montar o sistema de arquivos. O parâmetro `-t reiserfs` poderá ser omitido, se desejar.
4. Confira se o sistema de arquivos ReiserFS em `arquivo-reiserfs` foi realmente montado no sistema de arquivos digitando `df -T`. Para detalhes, veja [df, Seção 10.3](#).

Pronto! o que você gravar para `/mnt` será gravado dentro do arquivo `/tmp/arquivo-reiserfs`. Você poderá usar todos os recursos de um sistema de arquivos `reiserfs` como permissões de arquivos e diretórios, links simbólicos, etc.

O uso da opção `loop=/dev/loop1` permite que o dispositivo `/dev/loop1` seja associado ao arquivo `/arquivo-reiserfs` e assim permitir sua montagem e uso no sistema.

- Você poderá usar apenas `-o loop` com o comando `mount`, assim o `kernel` gerenciará automaticamente os dispositivos de `loop`.
- Caso faça isto manualmente, lembre-se de usar dispositivos `/dev/loop?` diferentes para cada arquivo que montar no sistema. Pois cada um faz referência a um único arquivo.

5.6.3 Nomeando uma partição de disco

O comando `e2label` é usado para esta função.

```
e2label [dispositivo] [nome]
```

Onde:

dispositivo

Partição que terá o nome modificado

nome

Nome que será dado a partição (máximo de 16 caracteres). Caso seja usado um nome de volume com espaços, ele deverá ser colocado entre "aspas".

Se não for especificado um nome, o nome atual da partição será mostrado. O nome da partição também pode ser visualizado através do comando `dumpe2fs` (veja [dumpe2fs, Seção 5.6.5](#)).

Exemplo: `e2label /dev/sda1 FocaLinux`, `e2label /dev/sda1 "Foca Linux"`

5.6.4 Criando o diretório especial `lost+found`

O utilitário `mklost+found` cria o diretório especial `lost+found` no diretório atual. O diretório `lost+found` é criado automaticamente após a formatação da partição com o `mkfs.ext2`, a função deste diretório é pré-alocar os blocos de arquivos/diretório durante a execução do programa `fsck.ext2` na recuperação de um sistema de arquivos (veja [Checagem dos sistemas de arquivos, Seção 26.1](#)). Isto garante que os blocos de disco não precisarão ser diretamente alocados durante a checagem.

`mklost+found`

OBS: Este comando só funciona em sistemas de arquivos `ext2`

Exemplo: `cd /tmp;mklost+found;ls -a`

5.6.5 `dumpe2fs`

Mostra detalhes sobre uma partição Linux.

`dumpe2fs [opções] [partição]`

Onde:

partição

Identificação da partição que será usada.

opções

`-b`

Mostra somente os blocos marcado como defeituosos no sistema de arquivos especificado.

Este comando lista diversas opções úteis do sistema de arquivos como o tipo do sistema de arquivos, características especiais, número de inodos, blocos livres, tamanho do bloco, intervalo entre checagens automáticas, etc.

Exemplo: `dumpe2fs /dev/sda1`, `dumpe2fs -b /dev/sda1`

5.6.6 Partição EXT2 ou Arquivo?

Criar uma partição EXT2 ou um arquivo usando o `loop`? Abaixo estão algumas considerações:

- A partição EXT2 é o método recomendado para a instalação do GNU/Linux.
- O desempenho da partição EXT2 é bem melhor se comparado ao arquivo porque é acessada diretamente pelo Kernel (SO).

- O arquivo `EXT2` é útil para guardarmos dados confidenciais em disquetes ou em qualquer outro lugar no sistema. Você pode perfeitamente gravar seus arquivos confidenciais em um arquivo chamado `libBlaBlaBla-2.0` no diretório `/lib` e ninguém nunca suspeitará deste arquivo (acho que não...). Também é possível criptografá-lo para que mesmo alguém descobrindo que aquilo não é uma lib, não poder abri-lo a não ser que tenha a senha (isto é coberto no documento `Loopback-encrypted-filessystem.HOWTO`).
 - O uso do arquivo `EXT2` é útil quando você está perdendo espaço na sua partição `EXT2` e não quer re-particionar seu disco pois teria que ser feita uma re-instalação completa e tem muito espaço em um partição de outro SO (como o Windows).
Você poderia facilmente copiar o conteúdo de `/var`, por exemplo, para o arquivo `EXT2` `ext2-1` criado no diretório Raiz do Windows, apagar o conteúdo de `/var` (liberando muito espaço em disco) e então montar `ext2-1` como `/var`. A partir de agora, tudo o que for gravado em `/var` será na realidade gravado no arquivo `ext2-1`.
Para o sistema acessar o arquivo, deve passar pelo sistema de arquivos `loop` e `FAT32`, isto causa um desempenho menor.
-

5.7 Partição Linux Swap (Memória Virtual)

Este tipo de partição é usado para oferecer o suporte a *memória virtual* ao GNU/Linux em adição a *memória RAM* instalada no sistema. Este tipo de partição é identificado pelo tipo 82 nos programas de particionamento de disco para Linux. Para detalhes de como criar uma partição Linux Swap veja [Criando sistema de arquivos Swap em uma partição, Seção 5.7.1](#).

Somente os dados na memória RAM são processados pelo processador, por ser mais rápida. Desta forma quando você está executando um programa e a memória RAM começa a encher, o GNU/Linux move automaticamente os dados que não estão sendo usados para a partição Swap e libera a memória RAM para a continuar carregando os dados necessários pelo. Quando os dados movidos para a partição Swap são solicitados, o GNU/Linux move os dados da partição Swap para a Memória. Por este motivo a partição Swap também é chamada de Troca ou memória virtual.

A velocidade em que os dados são movidos da memória RAM para a partição é muito alta. Note também que é possível criar o sistema de arquivos Swap em um arquivo ao invés de uma partição (veja [Criando um sistema de arquivos Swap em um arquivo, Seção 5.7.2](#)).

5.7.1 Criando sistema de arquivos Swap em uma partição

O programa usado para formatar uma partição Swap é o `mkswap`. Seu uso é simples:

```
mkswap /dev/hda?
```

Novamente veja [Identificação de discos e partições em sistemas Linux, Seção 5.12](#) caso não souber identificar seus discos e partições. O nome do dispositivo da partição Swap pode ser visualizado através de seu programa de particionamento, você pode usar o comando `fdisk -l /dev/hda` para listar as partições no primeiro disco rígido e assim verificar qual dispositivo corresponde a partição Swap.

A opção `-c` também pode ser usada com o `mkswap` para checar se existem agrupamentos danificados na partição.

Com a partição Swap formatada, use o comando: `swapon /dev/hda?` para ativar a partição Swap (lembre-se de substituir ? pelo número de sua partição Swap).

Observações:

Os Kernels do GNU/Linux 2.0.xx e anteriores somente suportam partições Swap de até 128MB. Caso precise de mais que isso, crie mais partições Swap ou atualize seu sistema para trabalhar com o kernel 2.2.xx

Se utilizar mais que 1 partição Swap, pode ser útil o uso da opção `-p NUM` que especifica a prioridade em que a partição Swap será usada. Pode ser usado um valor de prioridade entre 0 e 32767, partições com número maior serão usadas primeiro, sendo que na montagem automática através de "mount -a" podem ser designados números negativos.

Procure usar o número maior para partições mais rápidas (elas serão acessadas primeiro) e números maiores para partições mais lentas. Caso precise desativar a partição Swap, use o comando: `swapoff /dev/hda?`.

5.7.2 Criando um sistema de arquivos Swap em um arquivo

Também é possível criar um arquivo que poderá ser usado como memória virtual. Veja passo a passo como fazer isso:

1. Use o comando `dd if=/dev/zero of=/tmp/troca bs=1024 count=16000` para criar um arquivo chamado `troca` vazio de 16Mb de tamanho em `/tmp`. Você pode modificar os parâmetros de `of` para escolher onde o arquivo será criado, o tamanho do arquivo poderá ser modificado através de `count`.
2. Execute `mkswap /tmp/troca` para formatar o arquivo. Após concluir este passo, o sistema de arquivos Swap estará criado e pronto para ser usado.
3. Digite `sync` para sincronizar os buffers para o disco, assim você não terá problemas em um servidor com muito I/O.
4. Ative o arquivo de troca com o comando `swapon /tmp/troca`.
5. Confira se o tamanho da memória virtual foi modificado digitando `cat /proc/meminfo` ou `free`.

Observações:

- Podem ser usadas partições de troca e arquivos de troca juntos, sem problemas.
- Caso seu sistema já tenha uma partição de Swap, é recomendável deixar o acesso ao arquivo Swap com uma prioridade menor (usando a opção `-p NUM` com o comando `swapon`).

5.7.3 Partição Swap ou Arquivo?

Criar uma partição de Troca ou um arquivo de troca? Abaixo algumas vantagens e desvantagens:

- A partição Swap é mais rápida que o arquivo Swap pois é acessada diretamente pelo Kernel. Se o seu computador tem pouca memória (menos que 32Mb) ou você tem certeza que o sistema recorre freqüentemente a memória virtual para executar seus programas, é recomendável usar uma partição Swap.

- O arquivo de troca permite que você crie somente uma partição `Linux Native` e crie o arquivo de troca na partição `EXT2`.
 - Você pode alterar o tamanho do arquivo de troca facilmente apagando e criando um novo arquivo como descrito em [Criando um sistema de arquivos Swap em um arquivo, Seção 5.7.2](#).
 - É possível criar um arquivo de troca em outros tipos de partições como `FAT16`, `FAT32`, etc.
 - O arquivo de troca estará disponível somente após o sistema de arquivos que o armazena (`ext2`, `fat32`, etc) estar montado. Isto é um problema para sistemas com pouca memória que dependem do arquivo de troca desde sua inicialização.
-

5.8 O sistema de arquivos `/proc`

É o sistema de arquivos do Kernel do `GNU/Linux`. Ele oferece um método de ler, gravar e modificar dinamicamente os parâmetros do kernel, muito útil para curiosos (como eu) e programas de configuração. A modificação dos arquivos do diretório `/proc` é o método mais usado para modificar a configuração do sistema e muitos programas também dependem deste diretório para funcionar.

Nele você tem todo o controle do que o seu sistema operacional está fazendo, a configuração dos hardwares, interrupções, sistema de arquivos montado, execução de programas, memória do sistema, rede, etc.

Agora entre no diretório `/proc` digite `ls` e veja a quantidade de arquivos e diretórios que ele possui, dê uma passeada por eles. Abaixo a descrição de alguns deles (todos podem ser visualizados pelo comando `cat`):

- `Diretórios com números` - Estes identificam os parâmetros de um processo em execução. Por exemplo, se o PID (identificação do processo) do `inetd` for 115, você pode entrar no diretório 115 e verificar as opções usadas para execução deste programa através de cada arquivos existente dentro do diretório. Alguns são:
- `cmdline` - O que foi digitado para iniciar o processo (pode também ter sido iniciado através de um programa ou pelo kernel).
- `environ` - Variáveis de Ambiente existentes no momento da execução do processo.
- `status` - Dados sobre a execução do Processo (PID, status da execução do programa, memória consumida, memória executável, UID, GID, etc).
- `apm` - Dados sobre o gerenciamento de energia
- `cmdline` - Linha de comando usada para inicializar o Kernel `GNU/Linux`. Os parâmetros são passados através do programa de inicialização, como o `LILO`, `LOADLIN`, `SYSLINUX`.
- `cpuinfo` - Detalhes sobre a CPU do sistema
- `devices` - Dispositivos usados no sistema
- `dma` - Canais de DMA usados por dispositivos
- `filesystems` - Sistemas de arquivos em uso atualmente
- `interrupts` - Interrupções usadas por dispositivos
- `ioports` - Portas de Entrada e Saída usadas pelos dispositivos do sistema
- `kcore` - Este arquivo corresponde a toda a memória RAM em seu sistema. Seu tamanho é correspondente a memória RAM do micro
- `kmsg` - Permite visualizar mensagens do Kernel (use o comando `cat < kmsg` para visualiza-lo e pressione CTRL+C para cancelar
- `loadavg` - Média de Carga do sistema

- `meminfo` - Dados de utilização da memória do sistema
- `misc` - Outras configurações
- `modules` - Módulos atualmente carregados no kernel
- `mounts` - Sistemas de Arquivos atualmente montados
- `pci` - Detalhes sobre dispositivos PCI do sistema
- `rtc` - Relógio em Tempo real do sistema
- `uptime` - Tempo de execução do sistema
- `version` - Versão atual do Kernel, programa usado na compilação, etc
- Diretório `net` - Dados sobre a rede do sistema
- Diretório `sys` - Dados sobre outras áreas do sistema
- Diretório `scsi` - Detalhes sobre dispositivos SCSI do sistema

Note que o diretório `proc` e os arquivos existentes dentro dele estão localizados no diretório raiz (`/`), mas não ocupa nenhum espaço no disco rígido.

5.9 LVM - Logical Volume Manager

O `lvm` (*Logical Volume Manager*) faz a associação entre dispositivos/partições físicas (incluindo discos RAID, MD, mass storages diversos, MD, e loop) e dispositivos lógicos. O método tradicional faz a alocação de todo espaço físico ao tamanho da partição do disco (o método tradicional), o que traz muito trabalho quando o espaço esgota, cópia de dados ou planejamento de uso de máquina (que pode mudar com o passar do tempo). O sistema de `lvm` soluciona os seguintes problemas:

- Uso eficaz de disco, principalmente quando há pouco espaço para criação de partições independentes.
- Permite aumentar/diminuir dinamicamente o tamanho das partições sem reparticionamento do disco rígido usando o espaço livre em outras partições ou utilizando o espaço livre reservado para o uso do LVM.
- Uma partição de disco é identificada por um nome de volume e não pelo dispositivo. Você pode então se referir aos volumes como: utilizadores, vendas, diretoria, etc.
- Sua divisão em 3 camadas possibilita a adição/remoção de mais discos de um conjunto caso seja necessário mais espaço em volumes, etc.
- Permite selecionar o tamanho do cluster de armazenamento e a forma que eles são acessados entre os discos, possibilitando garantir a escolha da melhor opção dependendo da forma que os dados serão manipulados pelo servidor.
- Permite snapshots dos volumes do disco rígido.

As 3 camadas do LVM são agrupadas da seguinte forma:

- `PV` (`Physical Volume`) - Corresponde a todo o disco rígido/partição ou dispositivo de bloco que será adicionado ao LVM. Os aplicativos que manipulam o volume físico, começam com as letras `pv*`. O espaço disponível no PV é dividido em PE (`Physical Extends`, ou extensões físicas). O valor padrão do PE é de 4MB, possibilitando a criação de um VG de 256Gb.

Por exemplo: `/dev/hda1`

- **VG (Volume Group)** - Corresponde ao grupo de volumes físicos que fazem parte do LVM. Do grupo de volume são alocados os espaços para criação dos volumes lógicos. Os aplicativos que manipulam o o grupo de volume, começam com as letras `vg*`.

Por exemplo: `/dev/lvmdisk0` **LV (Logical Volume)** - Corresponde a partição lógica criada pelo LVM para gravação de dados. ao invés de ser identificada por nomes de dispositivos, podem ser usados nomes comuns para se referir as partições (`tmp`,`usr`,etc.). O Volume lógico é a área onde o sistema de arquivo é criado para gravação de dados, seria equivalente a partição em um sistema *SEM LVM* só que lógica ao invés de física. O volume lógico tem seu espaço dividido em LE (Logical Extends, ou extensões lógicas) que correspondem aos PE's alocados.

Exemplos: `/dev/lvmdisk/usr`, `/dev/lvmdisk/tmp`, etc.

5.9.1 Representação gráfica do LVM

Desenvolvi este desenho para representar a idéia de organização de um sistema LVM para o guia Foca GNU/Linux e apresentar a descrição prática da coisa:

```
+-----[ Grupo de Volume (VG) - lvmdisk ]-----+

|  +--[ PV - hda1 ]---+  +--[ PV - hdb1 ]--+  |
|
|  |  PE  PE  PE  PE  PE  PE  |  |  PE  PE  PE  PE  PE  |  |
|
|  +-----+  +-----+  |
|
|  |  |  |  |  |  |
|
|  |  |  +-----+  |  |
|
|  |  +-----+  |  |
|
|  |  |  |  |  |  |
|
|  +--[ LV - var ]--+  +--[ LV - home ]--+  |
|
|  |  LE  LE  LE  LE  |  |  LE  LE  LE  LE  |  |
|
|  +-----+  +-----+  |
+-----+
```

O gráfico acima representa a seguinte situação:

- Nós temos dois volumes físicos representados por `hda1` e `hdb1`. Cada um desses volumes físicos tem um Physical Extend (PE) de 4M (o padrão).
- Estes dois volumes físicos acima representam o espaço total do grupo de volume *lvmdisk* em `/dev/lvmdisk`.

- Do grupo de volume *lvmdisk* são criados dois volumes lógicos chamados *var* e *home*, estando disponíveis para particionamento através de `/dev/lvmdisk/var` e `/var/lvmdisk/home`.

Na prática, o espaço do volume lógico é definido alocando-se alguns Physical Extends (PE) dos volumes físicos como logical extends (LE) dos volumes lógicos. Desta forma, o tamanho de todos os PEs e LEs existentes dentro de um mesmo grupo de volume devem ser iguais.

5.9.2 Performance do LVM

Um sistema com LVM tem sua performance um pouco reduzida quanto ao acesso a disco, devido as camadas adicionais de acesso aos dados, sendo afetadas operações em caracteres e inteligentes de acesso a dados.

Entretanto, a performance de leitura/gravação de blocos é melhorada consideravelmente após a adoção do LVM. O LVM também garante que o sistema não mostre sintomas de paradas durante o esvaziamento de cache de disco, mantendo sempre uma certa constância na transferência de dados mesmo em operações pesadas de I/O no disco. Depende de você avaliar estes pontos e considerar sua adoção.

5.9.3 Colocando LVM em seu sistema

Nesta seção não tenho a intenção de cobrir todos os detalhes técnicos da implantação do LVM, a idéia aqui é fornecer uma referência básica e prática para uso em qualquer sistema normal (desconsiderando usos críticos). A idéia aqui é mostrar de forma prática como implantar LVM em sua máquina e preparar seu uso nos discos.

Antes de começar, retire QUALQUER CD que estiver inserido na unidade de CD-ROM, pois eles podem causar erro no `pvs`, `pvdisplay`, etc.

- No particionamento, defina as partições do tipo 8E (Linux LVM). A partição Linux LVM é exatamente igual a Linux Native (82), a única vantagem é que o LVM utilizará auto detecção para saber quais partições ele deve utilizar no `pvs`.

- Instale o pacote `lvml0` e uma imagem de kernel 2.4 que tenha suporte a LVM, ou compile seu próprio kernel (caso goste de máquinas turbinadas :-)

- Execute o `pvs` para detectar as partições marcadas como LVM e criar sua configuração em `/etc/lvmtab.d`.

OBS: É normal o sistema procurar dispositivos de CD-ROM durante a execução do `pvs`, apenas não deixe um CD na unidade para evitar grandes sustos se estiver desatento com os passos :-)

- Rode o `pvcreate` no disco ou partição para dizer que ela será um volume físico do LVM: `pvcreate /dev/hda1` ou `pvcreate /dev/hda`

Em caso de dúvida sobre qual é a partição LVM, digite: `fdisk -l /dev/hda` (supondo que `/dev/hda` é o disco rígido que está configurando o LVM).

- Rode o `pvdisplay /dev/hda1` para verificar se o volume físico foi criado. Recomendo que deixe a partição raiz (/) de fora do LVM para não ter futuros problemas com a manutenção do seu sistema, a menos que tenha muitas opções de inicialização com suporte a LVM em mãos, ou algo mais complexo baseado em `initrd` :-)

- Crie o grupo de volume na partição `vgcreate lvmdisk /dev/hda1 /dev/hdb7...` Note que partições de discos diferentes podem fazer parte de um mesmo grupo de volume (VG) do LVM. Caso use o `devfs`, será preciso usar o caminho completo do dispositivo ao invés do link: `vgcreate lvmdisk /dev/ide/host0/bus0/target0/lun0/part1`

O valor padrão do "Physical Extend" é de 4MB mas pode ser alterado pelo parâmetro "-s tamanho", assim o tamanho máximo do grupo de volume será de 256GB (4MB * 64.000 extends que são suportados por volume lógico). Os valores do Physical Extend (PE) pode ser de 8k a 16GB. Não é possível modificar o tamanho do PE após ele ser definido.

- Verifique o grupo de volume (VG) recém criado com o comando: `vgdisplay` ou `vgdisplay /dev/hda6`. Atente para a linha "Free PE / tamanho", que indica o espaço livre restante para criar os volumes lógicos (LV).
- Crie o volume lógico (LV) com o comando: `lvcreate -L1500 -ntmp lvmdisk` Que vai criar uma partição LVM de 1500MB (1,5GB) com o nome *tmp* (acessível por `/var/lvmdisk/tmp`) dentro do grupo *lvmdisk*. Você deverá fazer isso com as outras partições.
- Agora resta criar um sistema de arquivos (*ext3*, *reiserfs*, *xf*s, *jfs*, etc) como faria com qualquer partição física normal:

```
mkfs.ext3 /dev/lvmdisk/tmp mkfs.reiserfs /dev/lvmdisk/tmp
```

OBS: Caso deseje montar automaticamente o volume LVM, coloque o caminho completo do LVM ao invés do volume físico no `/etc/fstab:/dev/lvmdisk/tmp`.

5.9.4 Aumentando o tamanho de um volume lógico

O processo para aumentar o tamanho do volume lógico consiste em primeiro aumentar o tamanho do VG com o `lvextend` e depois ajustar o tamanho do sistema de arquivos:

```
# Aumenta o espaço do volume lógico tmp para 1G
```

```
lvextend -L1G /dev/lvmdisk/tmp
```

```
# Aumenta em 200MB o espaço no volume lógico tmp
```

```
lvextend -L+200M /dev/lvmdisk/tmp
```

As unidades `Kk`, `Mm`, `Gg`, `Tt` podem ser usadas para especificar o espaço. Após modificar o volume lógico, será preciso aumentar o tamanho do sistema de arquivos para ser exatamente igual ao tamanho do LV. Isto depende do seu sistema de arquivos:

```
ext2/3
```

```
resize2fs /dev/lvmdisk/tmp
```

O `ext2/3` ainda vem com o utilitário `e2fsadm` que executa os dois comandos (`lvextend` e `resize2fs`) de uma só vez: `e2fsadm -L+1G /dev/lvmdisk/tmp`

OBS: Você deverá desmontar o sistema de arquivos antes de alterar o tamanho de um sistema de arquivos *ext2* ou *ext3*. Para alterar o tamanho durante a execução do sistema operacional, é necessária a aplicação do patch *ext2online* no kernel.

```
reiserfs
```

```
resize_reiserfs -f /dev/lvmdisk/tmp
```

O tamanho do sistema de arquivos `reiserfs` poderá ser modificado on-line, assim não precisa parar seu servidor para esta operação.

xfs

xfs_growfs /tmp

Note que deve ser especificado o ponto de montagem ao invés do dispositivo. O sistema de arquivos deverá ser montado antes de ser modificado e incluído no `/etc/fstab`.

5.9.5 Diminuindo um volume lógico

Para diminuir o tamanho de um volume lógico, certifique-se de ter calculado o espaço corretamente para acomodar todos os dados que já existem na partição. A diferença para o processo de aumentar o LV é que neste o sistema de arquivos é reduzido primeiro e depois o LV:

ext2/3

```
e2fsadm -L-1G /dev/lvmdisk/tmp
```

Você também poderá usar o `resize2fs` e depois o `lvreduce`, mas deverá dizer o tamanho em blocos para o `resize2fs` que varia de acordo com o tamanho do sistema de arquivos:

```
resize2fs /dev/lvmdisk/tmp 524288
```

```
lvreduce -L-1G /dev/lvmdisk/tmp
```

OBS: Você deverá desmontar o sistema de arquivos antes de alterar o tamanho do sistema de arquivos, a não ser que tenha o patch *ext2online* aplicado no kernel.

reiserfs

```
resize_reiserfs -s-1G /dev/lvmdisk/tmp lvreduce -L-1G /dev/lvmdisk/tmp
```

O tamanho do sistema de arquivos `reiserfs` poderá ser modificado on-line, assim não precisa parar seu servidor para a modificação.

xfs

Não é possível diminuir o tamanho de um sistema de arquivos XFS em sua versão atual (12/2003).

5.10 Formatando disquetes

As subseções seguintes explicarão maneiras de formatar seus disquetes para serem usados no GNU/Linux e DOS/Windows.

5.10.1 Formatando disquetes para serem usados no Linux

Para formatar disquetes para serem usados no GNU/Linux use o comando:

```
mkfs.ext2 [-c] [/dev/fd0]
```

Em alguns sistemas você deve usar `mke2fs` no lugar de `mkfs.ext2`. A opção `-c` faz com que o `mkfs.ext2` procure por blocos danificados no disquete e `/dev/fd0` especifica a primeira unidade de disquetes para ser formatada (equivalente a A: no DOS). Mude para `/dev/fd1` para formatar um disquete da segunda unidade.

OBS: Este comando cria um sistema de arquivos `ext2` no disquete que é nativo do GNU/Linux e permite usar características como permissões de acesso e outras. Isto também faz com que o disquete NÃO possa ser lido pelo DOS/Windows. Para formatar um disquete no GNU/Linux usando o `FAT12` (compatível com o DOS/Windows) veja próxima seção.

Exemplo: `mkfs.ext2 -c /dev/fd0`

5.10.2 Formatando disquetes compatíveis com o DOS/Windows

A formatação de disquetes DOS no GNU/Linux é feita usando o comando `superformat` que é geralmente incluído no pacote `mttools`. O `superformat` cria um sistema de arquivos em um disquete para ser usado no DOS e também possui opções avançadas para a manipulação da unidade, formatação de intervalos de cilindros específicos, formatação de discos em alta capacidade e verificação do disquete.

`superformat [opções] [dispositivo]`

dispositivo

Unidade de disquete que será formatada. Normalmente `/dev/fd0` ou `/dev/fd1` especificando respectivamente a primeira e segunda unidade de disquetes.

opções

`-v [num]`

Especifica o nível de detalhes que serão exibidos durante a formatação do disquete. O nível 1 especifica um ponto mostrado na tela para cada trilha formatada. Veja a página de manual do `superformat` para detalhes.

`-sverify`

Verifica primeiro se a trilha pode ser lida antes de formatá-la. Este é o padrão.

`--dosverify, -B`

Verifica o disquete usando o utilitário `mbadblocks`. Usando esta opção, as trilhas defeituosas encontradas serão automaticamente marcadas para não serem utilizadas.

`--verify_later, -V`

Verifica todo o disquete no final da formatação.

`--noverify, -f`

Não faz verificação de leitura.

`-b [trilha]`

Especifica a trilha inicial que será formatada. O padrão é 0.

`-e [trilha]`

Especifica a trilha final que será formatada.

Na primeira vez que o `superformat` é executado, ele verifica a velocidade de rotação da unidade e a comunicação com a placa controladora, pois os discos de alta densidade são sensíveis a rotação da unidade. Após o teste inicial ele recomendará adicionar uma linha no arquivo `/etc/driveprm` como forma de evitar que este teste seja sempre executado. *OBS:* Esta linha é calculada de acordo com a rotação de uma unidade de disquetes, transferência de dados e comunicação com a placa controladora de disquete. Desta forma ela varia de computador para computador. Note que não é necessário montar a unidade de disquetes para formatá-la. Segue abaixo exemplos de como formatar seus disquetes com o `superformat`:

- `superformat /dev/fd0` - Formata o disquete na primeira unidade de disquetes usando os valores padrões.
 - `superformat /dev/fd0 dd` - Faz a mesma coisa que o acima, mas assume que o disquete é de Dupla Densidade (720Kb).
 - `superformat -v 1 /dev/fd0` - Faz a formatação da primeira unidade de disquetes (`/dev/fd0`) e especifica o nível de detalhes para 1, exibindo um ponto após cada trilha formatada.
-

5.10.3 Programas de Formatação Gráficos

Além de programas de formatação em modo texto, existem outros para ambiente gráfico (X11) que permitem fazer a mesma tarefa.

Entre os diversos programas destaco o `gfloppy` que além de permitir selecionar se o disquete será formatado para o GNU/Linux (ext2) ou DOS (FAT12), permite selecionar a capacidade da unidade de disquetes e formatação rápida do disco.

5.11 Pontos de Montagem

O GNU/Linux acessa as partições existente em seus discos rígidos e disquetes através de diretórios. Os diretórios que são usados para acessar (montar) partições são chamados de *Pontos de Montagem*. Para detalhes sobre montagem de partições, veja [Montando \(acessando\) uma partição de disco, Seção 5.13](#).

No DOS cada letra de unidade (C:, D:, E:) identifica uma partição de disco, no GNU/Linux os pontos de montagem fazem parte da grande estrutura do sistema de arquivos raiz.

Existem muitas vantagens de se usar *pontos de montagem* ao invés de unidade de disco para identificar partições (método usado no DOS):

- Você pode montar a partição no diretório que quiser.
- Em caso de um sistema de arquivos cheio, você pode copiar o conteúdo de um grande diretório para um disco separado, apagar o conteúdo do diretório original e montar o disco onde foram copiados os arquivos naquele local (caso não use um sistema de LVM).
- O uso de *pontos de montagem* torna o gerenciamento mais flexível.
- A adição de novas partições ou substituição de discos rígidos não afeta a ordem de identificação dos discos e pontos de montagem (como não acontece no DOS).

5.12 Identificação de discos e partições em sistemas Linux

No GNU/Linux, os dispositivos existentes em seu computador (como discos rígidos, disquetes, tela, portas de impressora, modem, etc) são identificados por um arquivo referente a este dispositivo no diretório `/dev`.

A identificação de discos rígidos no GNU/Linux é feita da seguinte forma:

`/dev/hda1`

| | |

| | | _Número que identifica o número da partição no disco rígido.

| | |

| | | _Letra que identifica o disco rígido (a=primeiro, b=segundo, etc...).

| |

| | _Sigla que identifica o tipo do disco rígido (hd=ide, sd=SCSI, xt=XT).

|

| _Diretório onde são armazenados os dispositivos existentes no sistema.

Abaixo algumas identificações de discos e partições em sistemas Linux:

- `/dev/fd0` - Primeira unidade de disquetes.
- `/dev/fd1` - Segunda unidade de disquetes.
- `/dev/hda` - Primeiro disco rígido na primeira controladora IDE do micro (primary master).
- `/dev/hda1` - Primeira partição do primeiro disco rígido IDE.
- `/dev/hdb` - Segundo disco rígido na primeira controladora IDE do micro (primary slave).
- `/dev/hdb1` - Primeira partição do segundo disco rígido IDE.
- `/dev/sda` - Primeiro disco rígido na primeira controladora SCSI ou SATA.
- `/dev/sda1` - Primeira partição do primeiro disco rígido SCSI ou SATA.
- `/dev/sdb` - Segundo disco rígido na primeira controladora SCSI ou SATA.
- `/dev/sdb1` - Primeira partição do segundo disco rígido SCSI ou SATA.
- `/dev/sr0` - Primeiro CD-ROM SCSI.
- `/dev/sr1` - Segundo CD-ROM SCSI.
- `/dev/xda` - Primeiro disco rígido XT.
- `/dev/xdb` - Segundo disco rígido XT.

As letras de identificação de discos rígidos podem ir além de `hdb`, em meu micro, por exemplo, a unidade de CD-ROM está localizada em `/dev/hdg` (Primeiro disco - quarta controladora IDE).

É importante entender como os discos e partições são identificados no sistema, pois será necessário usar os parâmetros corretos para monta-los.

5.13 Montando (acessando) uma partição de disco

Você pode acessar uma partição de disco usando o comando `mount`.

```
mount [dispositivo] [ponto de montagem] [opções]
```

Onde:

dispositivo

Identificação da unidade de disco/partição que deseja acessar (como `/dev/hda1` (disco rígido) ou `/dev/fd0` (primeira unidade de disquetes).

ponto de montagem

Diretório de onde a *unidade de disco/partição* será acessado. O diretório deve estar vazio para montagem de um sistema de arquivo. Normalmente é usado o diretório `/mnt` para armazenamento de pontos de montagem temporários.

`-t [tipo]`

Tipo do sistema de arquivos usado pelo *dispositivo*. São aceitos os sistemas de arquivos:

- `ext2` - Para partições GNU/Linux usando o Extended File System versão 2 (a mais comum).
- `ext3` - Para partições GNU/Linux usando o Extended File System versão 3, com suporte a journaling.
- `reiserfs` - Para partições reiserfs, com suporte a journaling.
- `xfs` - Para partições xfs, com suporte a journaling.
- `vfat` - Para partições Windows 95 que utilizam nomes extensos de arquivos e diretórios.
- `msdos` - Para partições DOS normais.
- `iso9660` - Para montar unidades de CD-ROM. É o padrão.
- `umsdos` - Para montar uma partição DOS com recursos de partições EXT2, como permissões de acesso, links, etc.

Para mais detalhes sobre opções usadas com cada sistema de arquivos, veja a página de manual *mount*.

`-r`

Caso for especificada, monta a partição somente para leitura.

`-w`

Caso for especificada, monta a partição como leitura/gravação. É o padrão.

Existem muitas outras opções que podem ser usadas com o comando `mount`, mas aqui procurei somente mostrar o básico para "montar" seus discos e partições no GNU/Linux (para mais opções, veja a página de manual do `mount`). Caso você digitar `mount` sem parâmetros, serão mostrados os sistemas de arquivos atualmente montados no sistema. Esta mesma listagem pode ser vista em `/etc/mtab`. A remontagem de partição também é muito útil, especialmente após reparos nos sistema de arquivos do disco rígido. Veja alguns exemplos de remontagem abaixo.

É necessário permissões de root para montar partições, a não ser que tenha especificado a opção `user` no arquivo `/etc/fstab` (veja [fstab, Seção 5.13.1](#)).

Exemplo de Montagem:

- Montar uma partição Windows (vfat) de /dev/hda1 em /mnt somente para leitura: `mount /dev/hda1 /mnt -r -t vfat`
- Montar a primeira unidade de disquetes /dev/fd0 em /floppy: `mount /dev/fd0 /floppy -t vfat`
- Montar uma partição DOS localizada em um segundo disco rígido /dev/hdb1 em /mnt: `mount /dev/hdb1 /mnt -t msdos.`
- Remontar a partição raíz como somente leitura: `mount -o remount,rw /`
- Remontar a partição raíz como *leitura/gravação* (a opção -n é usada porque o mount não conseguirá atualizar o arquivo /etc/mtab devido ao sistema de arquivos / estar montado como somente leitura atualmente: `mount -n -o remount,rw /`.

5.13.1 fstab

O arquivo /etc/fstab permite que as partições do sistema sejam montadas facilmente especificando somente o dispositivo ou o ponto de montagem. Este arquivo contém parâmetros sobre as partições que são lidos pelo comando mount. Cada linha deste arquivo contém a partição que desejamos montar, o ponto de montagem, o sistema de arquivos usado pela partição e outras opções. fstab tem a seguinte forma:

Sistema_de_arquivos Ponto_de_Montagem Tipo Opções dump ordem

/dev/hda1 / ext2 defaults 0 1

/dev/hda2 /boot ext2 defaults 0 2

/dev/hda3 /dos msdos defaults,noauto,rw 0 0

/dev/hdg /cdrom iso9660 defaults,noauto 0 0

Onde:

Sistema de Arquivos

Partição que deseja montar.

Ponto de montagem

Diretório do GNU/Linux onde a partição montada será acessada.

Tipo

Tipo de sistema de arquivos usado na partição que será montada. Para partições GNU/Linux use *ext2*, para partições DOS (sem nomes extensos de arquivos) use *msdos*, para partições Win 95 (com suporte a nomes extensos de arquivos) use *vfat*, para unidades de CD-ROM use *iso9660*.

Opções

Especifica as opções usadas com o sistema de arquivos. Abaixo, algumas opções de montagem para ext2/3 (a lista completa pode ser encontrada na página de manual do mount):

- *defaults* - Utiliza valores padrões de montagem.
- *noauto* - Não monta os sistemas de arquivos durante a inicialização (útil para CD-ROMS e disquetes).
- *ro* - Monta como somente leitura.

- `user` - Permite que utilizadores montem o sistema de arquivos (não recomendado por motivos de segurança).
- `sync` é recomendado para uso com discos removíveis (disquetes, zip drives, etc) para que os dados sejam gravados imediatamente na unidade (caso não seja usada, você deve usar o comando [sync, Seção 10.22](#) antes de retirar o disquete da unidade).

`dump`

Especifica a frequência de backup feita com o programa `dump` no sistema de arquivos. 0 desativa o backup.

`Ordem`

Define a ordem que os sistemas de arquivos serão verificados na inicialização do sistema. Se usar 0, o sistema de arquivos não é verificado. O sistema de arquivos raiz que deverá ser verificado primeiro é o raiz "/" (a não ser que você tenha um sistema de arquivos de outro tipo que não é montado dentro do diretório raiz e possui seu suporte embutido no kernel) .

Após configurar o `/etc/fstab`, basta digitar o comando `mount /dev/hdg` ou `mount /cdrom` para que a unidade de CD-ROM seja montada. Você deve ter notado que não é necessário especificar o sistema de arquivos da partição pois o `mount` verificará se ele já existe no `/etc/fstab` e caso existir, usará as opções especificadas neste arquivo. Para maiores detalhes veja as páginas de manual `fstab` e `mount`.

5.14 Desmontando uma partição de disco

Para desmontar um sistema de arquivos montado com o comando `mount`, use o comando `umount`. Você deve ter permissões de root para desmontar uma partição.

```
umount [dispositivo/ponto de montagem]
```

Você pode tanto usar `umount /dev/hda1` como `umount /mnt` para desmontar um sistema de arquivos `/dev/hda1` montado em `/mnt`.

Observação: O comando `umount` executa o `sync` automaticamente no momento da desmontagem para garantir que todos os dados ainda não gravados serão salvos.