

Capítulo 5 - Dispositivos de bloco

5.1. Objetivos

- Entender o funcionamento do /dev;
- Diferenciar devfs de udev;
- Compreender o processo de montagem de um dispositivo;
- Testar alguns dispositivos presentes em /dev
- Compreender como funciona o particionamento;
- Aplicar um sistema de arquivos a uma partição.

5.2. Dispositivos em Linux

O núcleo do sistema operacional GNU/Linux, o “kernel”, se comunica com os dispositivos de uma maneira muito interessante: praticamente todos os dispositivos em GNU/Linux são representados por um arquivo correspondente dentro do sistema de arquivos. Exceções a esta regra são as placas de rede.

Alguns assuntos sobre dispositivos foram retirados da apostila pois estão ligeiramente ultrapassados, porém os mesmo podem ser encontrados no Anexo.

O local onde são armazenadas estas representações é o diretório `/dev`. Uma listagem deste diretório mostrará uma série de arquivos, todos eles representando uma parte do seu computador. A interação com estes arquivos pelo sistema operacional GNU/Linux realiza as leituras dos pedidos, processa estes pedidos e retorna as respostas controlando os sinais enviados aos dispositivos, por exemplo, comandando a placa de vídeo para coordenar as respostas no seu monitor.

5.2.1. Explorando o `/dev`

Uma diferença marcante entre sistemas Windows e Unix-like é a forma de lidar com partições e dispositivos como unidade de disquete e CD-ROM. Em sistemas Windows desde uma partição no disco rígido a um pen drive o acesso a eles são efetuados utilizando a idéia de “unidades” ou drives, como o drive C: ou A: ou até mesmo uma unidade de rede. Esse tipo de conceito faz com que o usuário final não precise saber o que está por detrás desses equipamentos, simplificando sua utilização ao preço da perda do conhecimento.

Em sistemas como GNU/Linux existe o conceito de dispositivos; praticamente tudo na máquina é tratado como sendo um dispositivo e pode ser acessado pelo seu respectivo arquivo localizado no diretório `/dev`. Uma exceção a isso é a interface de rede que é tratada diretamente no nível do kernel, não existindo um dispositivo (no `/dev`) associado a ela.

O diretório /dev consiste de um filesystem especial e pode ser de dois tipos: devfs ou udev. O devfs é o mais antigo tendo sido substituído pelo udev a partir do kernel 2.6.12. Uma das diferenças entre os dois é que no devfs os arquivos de dispositivos são criados uma única vez, dessa forma, o diretório /dev contém os dispositivos para todos os hardwares suportados pelo Linux, não importando se eles estão disponíveis na máquina ou não. Com o udev os dispositivos são criados de acordo com a disponibilidade no sistema. Dessa forma, o diretório contém apenas os arquivos de dispositivo para os hardwares presentes na máquina.



O UDEV não super popula o diretório dev do nosso sistema, além de nos proporcionar um método de configuração que pode ser encontrado em /etc/udev/

Explorando o diretório /dev você irá se deparar com alguns tipos de arquivos especiais, conhecidos como arquivos de dispositivos. Os tipos existentes são os dispositivos de:

- bloco** - utilizados para transferência de dados para hardwares de armazenamento de dados como discos rígidos;
- caracter** - conhecido também como "unbuffered" é utilizado para comunicação com hardwares como mice e terminais;
- fifo** - conhecido também como pipe nomeado (named pipe) é um dispositivo utilizado para realizar a comunicação entre dois processos independentes;
- socket** - um dispositivo do tipo socket é utilizado para criar um ponto de comunicação.

Seguindo essa classificação, os dois tipos mais comuns de serem manipulados são os de bloco e de caracter; como exemplos deles temos os devices referentes a dispositivos IDE conectados à máquina (/dev/hda1, por exemplo) e o dispositivo de acesso ao mouse (/dev/psaux, por exemplo).

Outros dispositivos de bloco importantes são os SCSI utilizados não apenas por discos SCSI mas também por dispositivos como USB e SATA, uma vez que são acessados utilizando essa emulação. O nome destes dispositivos são do tipo `/dev/sd[letra][número]` e seguem a mesma lógica dos dispositivos IDE. Dessa forma, se houver um HD SATA conectado à máquina e mais nenhum outro dispositivo que utilize emulação SCSI, sua localização será o device `/dev/sda`.

Os nomes dos dispositivos e a maneira como são representados na hierarquia do diretório `/dev` podem ser bastante difíceis a primeira vista. Com um pouco de prática, a nomenclatura usada fará sentido.

Um mouse USB é representado pelo arquivo `/dev/input/mice`, que pode ser traduzido como: dispositivo (DEV) de entrada (INPUT) do tipo apontador (MICE outro termo para “rato”, em inglês). Um mouse PS/2 segue uma nomenclatura um pouco mais complicada e é representada pelo arquivo `/dev/psaux`, que pode ser interpretado como dispositivo auxiliar na porta PS.

Para alguns dispositivos como o mouse podemos realmente ver a interação com o arquivo que representa o dispositivo. No exemplo abaixo, usamos o comando `cat` para mostrar o conteúdo do arquivo de dispositivo de mouse (mexa o mouse depois de pressionar **ENTER** após os comandos abaixo):

Para mice USB:



```
#cat /dev/input/mice
```

Para mice PS/2:



```
#cat /dev/psaux
```

As saídas, ilegíveis para humanos, representam os dados que o sistema operacional GNU/Linux usa para avaliar a movimentação, posicionamento e apertar de botões do mouse. Dispositivos de armazenamento

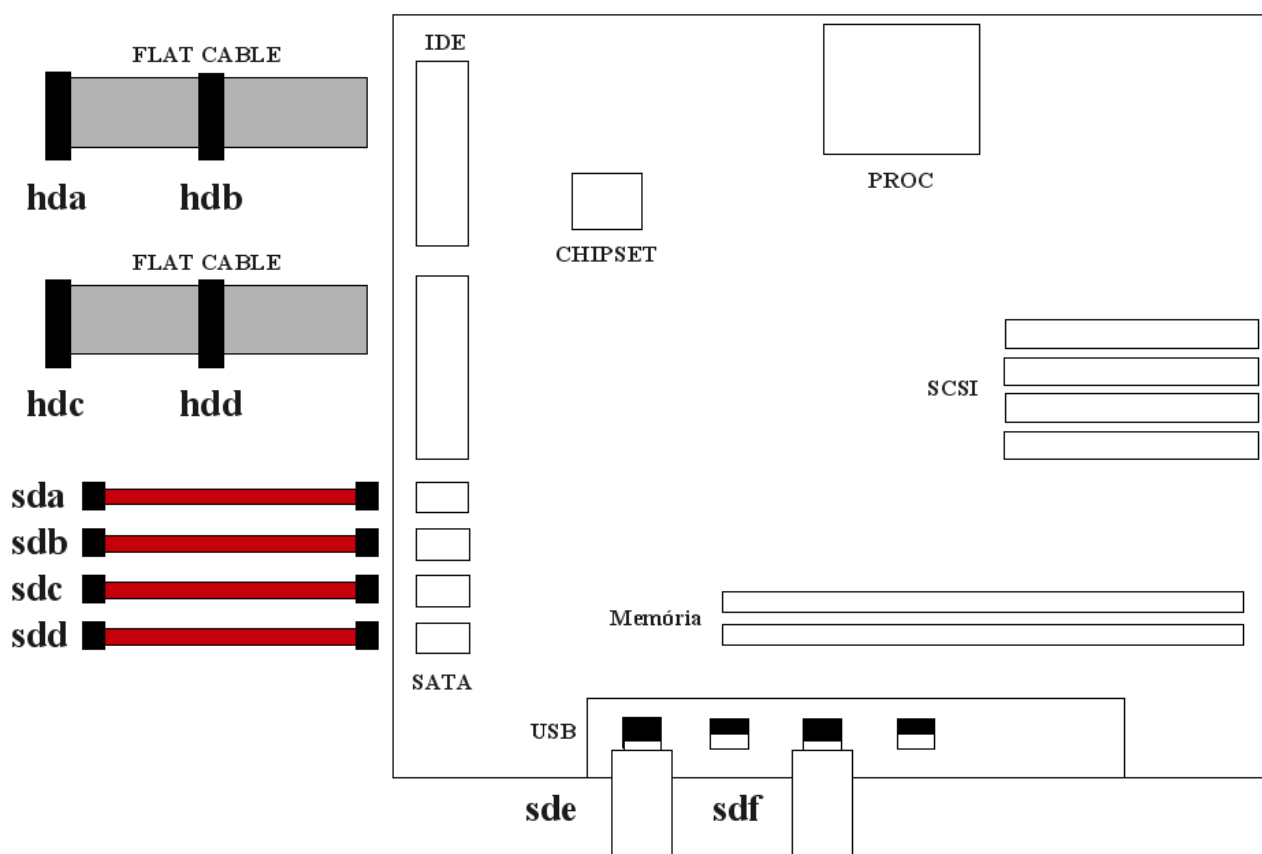


Ilustração 3: Placa mãe

Outro exemplo importante são os dispositivos de armazenamento principais do seu computador, os discos rígidos. Existem três tecnologias principais de discos rígidos, IDE, SATA e SCSI.

Os discos IDE ainda são maioria no mercado, mas a tecnologia vem dando lugar ao padrão SATA. Tanto o padrão IDE como o SATA são considerados econômicos e mais voltados para computadores pessoais ou estações de trabalho.

O discos do padrão SCSI usam uma tecnologia de acesso mais sofisticada, são geralmente mais rápidos que similares IDE e SATA e mais robustos. São usados principalmente em servidores e máquinas de alto desempenho.

Os dispositivos IDE são representados na hierarquia do diretório /dev com um padrão que começa com hd. O disco rígido conectado como mestre na controladora principal será designado por hda. Já o escravo, nesta mesma controladora, será representado por hdb. Analogamente, temos hdc e hdd respectivamente para os discos mestre e escravo conectados na controladora secundária.

Por outro lado, o padrão dos dispositivos SATA e SCSI começam por sd. Assim sendo, temos sda para o primeiro dispositivo SATA ou SCSI, sdb para o segundo, etc. Uma controladora SCSI de 8 bits pode comportar até 7 dispositivos, além da própria controladora. Para as de 16 bits, o número máximo de dispositivos é 15.

Podemos verificar o conteúdo de um disco usando novamente o comando cat. Para inspecionar o conteúdo do primeiro disco rígido IDE de um computador, podemos usar o comando abaixo:



```
# cat /dev/hda
```

A saída gerada não parece ter nenhum sentido. Os dados mostrados são aqueles dados gravados no seu disco. Contudo, estão em uma forma que é compreensível apenas pelo sistema operacional.

Uma partição é uma divisão lógica do seu disco rígido, criada por questões de organização, conveniência, flexibilidade ou segurança. Nos sistemas baseados em representação por letras, um disco rígido IDE pode ser dividido, particionado de forma a ser visto com as letras C: e D:. No sistema operacional GNU/Linux, esta mesma divisão levaria aos arquivos representados em /dev/hda1 e /dev/hda2. Ou seja, a primeira partição do disco hda é representada por /dev/hda1 e a segunda é representada por /dev/hda2. Qualquer partição adicional seguiria o mesmo padrão.

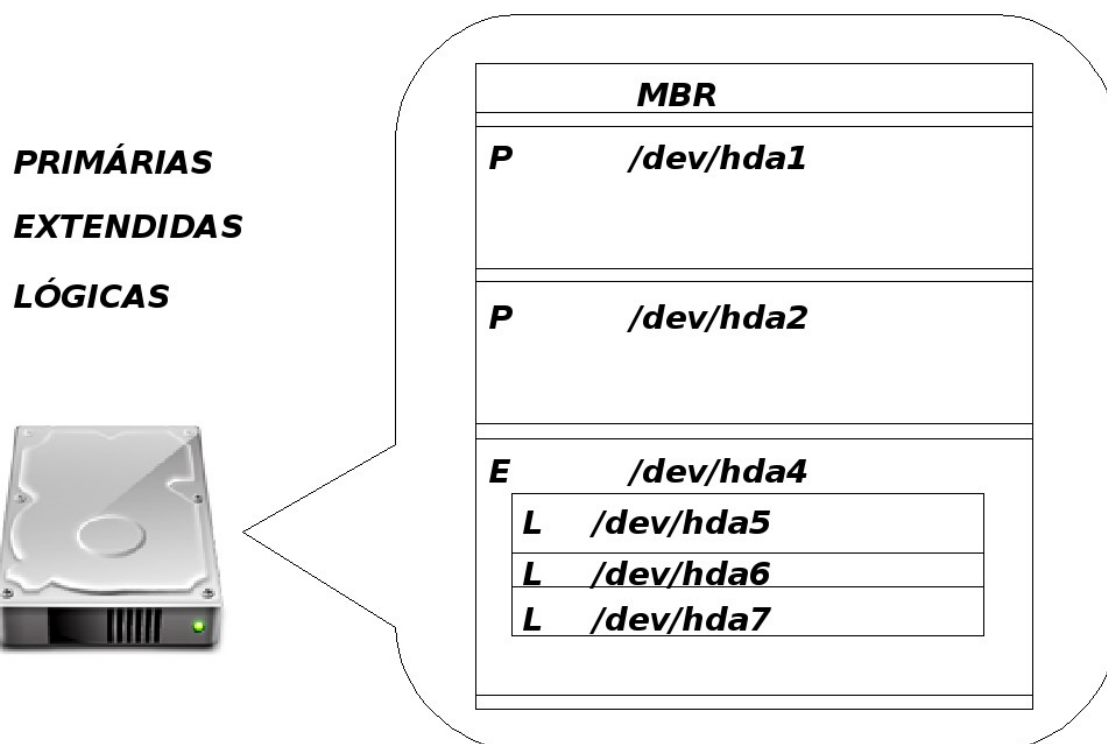


Ilustração 4: Estrutura das Partições

Assim, para inspecionar o conteúdo da primeira partição, pode-se usar o comando abaixo:



```
# cat /dev/hda1
```

Para interromper a saída do comando que pode ser bastante demorada, pressione a combinação de teclas Ctrl C (mantenha a tecla Ctrl pressionada e pressione a letra C). Caso a tela do seu console continue a mostrar caracteres estranhos, digite reset.

O último comando mostra uma saída que seres humanos não conseguem entender. Elas representam a maneira como os dados foram armazenados em /dev/hda1. Para que o sistema operacional GNU/Linux apresente estes dados de uma forma mais legível, é necessário solicitar ao sistema um processo de tradução. Este processo é chamado de montagem de dispositivos.

Então para que a partição `/dev/hda1` seja usada, é necessário montar esta partição em algum local e acessá-lo. Este local, que é um diretório no sistema de arquivos, é chamado de ponto de montagem. Podemos montar um dispositivo de armazenamento em qualquer diretório do sistema de arquivos, contudo, existem algumas convenções:

Dispositivos removíveis devem ser montados em `/media` (em outras épocas em `/mnt`).

Exemplos:

- **Um cdrom convencional**, representado por `/dev/cdrom` ou `/dev/hdc`, pode ser montado em `/media/cdrom`.

- **Um leitor de disquetes**, representado por `/dev/fd0`, pode ser montado em `/media/floppy`.

- **A grande maioria dos dispositivos de bloco USB**, são reconhecidos como `scsi`, e podem localizados em `/dev/sda`.

- **Um Hd Sata** também pode ser encontrado em `/dev/sda`, isso pode variar, depende da porta sata utilizada.

No caso de discos rígidos, uma partição é montada diretamente na raiz do sistema de arquivos ou em um diretório diretamente abaixo da raiz.

5.1. Devices, UUID e Labels

Quando usamos dispositivos seguindo padrões como `/dev/hda3` ou `/dev/sda5`, estamos especificando um dispositivo que pode vir a receber outro nome se houver alguma modificação no disco, isso implica no sistema não mais encontrar a partição especificada pois seu nome foi modificado. Uma alternativa inteligente para isso é utilizar o método UUID - Universally Unique Identifier ou utilizar o método de Labels.

Para descobrirmos o UUID de nossas partições podemos utilizar dois aplicativos: `vol_id` e `blkid`



```
# vol_id -u /dev/sda2  
f541a97e-ef19-4e47-b305-b535a75c932a
```


A **flag u** do comando `vol_id`, nos imprime a UUID referente a uma determinada partição.



```
# blkid
/dev/sda1: UUID="f541a97e-ef19-4e47-b305-b535a75c932a" TYPE="ext3"
LABEL="MAIN"
/dev/sda3: UUID="7C444A56444A12F6" TYPE="ntfs" LABEL="WIN"
/dev/sda5: TYPE="swap"
/dev/sda6: UUID="69ff8ed5-c09b-49b6-b21d-328e90243efa" TYPE="ext3"
LABEL="HOME"
/dev/sda7: UUID="2c070d34-5c6e-4504-8d4b-9a8fa910548d" TYPE="ext3"
LABEL="STORAGE"
/dev/sda8: UUID="489B-5A22" TYPE="vfat" LABEL="CENTER"
```

Já o comando `blkid` lista todos os dados relevantes sobre as partições do seu disco.

Há também um outro método de se descobrir essas informações para isso:



```
# ls -l /dev/disk/by-uuid/
lrwxrwxrwx 1 root root 10 2009-03-06 10:41 2c070d34-5c6e-4504-8d4b-
9a8fa910548d -> ../../sda7
lrwxrwxrwx 1 root root 10 2009-03-06 10:41 489B-5A22 -> ../../sda8
lrwxrwxrwx 1 root root 10 2009-03-06 10:41 69ff8ed5-c09b-49b6-b21d-
328e90243efa -> ../../sda6
lrwxrwxrwx 1 root root 10 2009-03-06 10:41 7C444A56444A12F6 ->
../../sda3
lrwxrwxrwx 1 root root 10 2009-03-06 10:41 f541a97e-ef19-4e47-b305-
b535a75c932a -> ../../sda1
```

Mas a resposta gerada não está tão amigável quando as outras.



O uso dos métodos de LABEL ou UUID em conjunto com o `/etc/fstab` nos proporcionam uma solução inteligente para o dia-a-dia e para nossa prova.

5.1.1. Usando os dispositivos de armazenamento

Para termos acesso a um arquivo armazenado em mídia removível, é necessário conectar a mídia removível ao seu leitor correspondente e montar o dispositivo adequado.

O comando usado para montar dispositivos é o mount. Sem o uso de nenhum parâmetro, ele mostra os dispositivos de armazenamento que estão montados em seu computador junto com a configuração usada para montá-los.



```
# mount
```



Existem muitos comandos para descobrirmos o que temos conectados em nossas máquinas, dentre eles: lspci, lsusb e lsscsi

Para montar um dispositivo de armazenamento em seu ponto de montagem, o comando mount pode ser usado da seguinte forma:



```
# mount -t <tipo> -o <opções> <dispositivo> <ponto-de-montagem>
```

Para que seja possível acessar o conteúdo de algum dispositivo precisamos de quatro itens básicos:

- saber qual o nome do dispositivo que será acessado;
- saber qual o filesystem que ele está utilizando;
- ter um ponto de montagem;
- ter permissão de montagem;

O método mais garantido de encontrar o nome de um dispositivo é realizar uma busca na saída do comando dmesg; por exemplo, se desejarmos determinar qual o nome do device do CD-ROM, podemos tentar:



```
# dmesg |grep ATAPI
```



As informações providas pelo comando dmesg são providas pelo arquivo /var/log/messages

Uma vez determinado o nome do dispositivo podemos, realizar outra procura no dmesg, mas agora com o nome do dispositivo, e determinar qual filesystem está utilizando.

Se não existir um ponto de montagem, basta criar um diretório vazio no local apropriado; em geral no /media ou /mnt e executar o comando para montá-lo. Por exemplo:

Para um cdrom, a sintaxe do comando seria:



```
# mount -t iso9660 /dev/cdrom /media/cdrom
```

Na maioria dos ambientes gráficos, este processo de montagem é automatizado. A simples inserção ou conexão de mídias removíveis faz com que elas sejam montadas e acessíveis pelos navegadores de arquivos gráficos.

Para desmontar um dispositivo, o comando usado é o umount. Neste caso é possível usar como parâmetro o ponto de montagem ou o dispositivo:

Por exemplo:



```
# umount /media/cdrom
```

ou de forma equivalente:



```
# umount /dev/cdrom
```



Uma alternativa para a montagem aleatória do sistema, é utilizar o pacote autofs que nos prove toda uma estrutura configurável para os dispositivos. Para usar autofs seu sistema precisa usar Kernel 2.6 e ter a partição /dev com udev

5.1. Criando Partições no HD

Agora que já sabemos como montar um dispositivo precisamos saber como criar uma partição manualmente. Para isso, há duas ferramentas importantes disponíveis em sistemas GNU/Linux, são elas fdisk e cfdisk.



Conhecer esses particionadores é muito importante, anote mais uma ai: Gparted

5.1.1. Particionamento com FDISK

O particionador fdisk é o mais completo dos particionadores e que em geral resolve nossos problemas quando eles ocorrem.

Fazendo a chamada a esse programa podemos ver a seguinte tela inicial:



```
# fdisk /dev/hda
```

*The number of cylinders for this disk is set to 14593.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)*

Command (m for help):

Pressionando a tecla **m** para obtermos um help, veremos a seguinte saída:



Command (m for help): m

Command action

<i>a</i>	<i>toggle a bootable flag</i>
<i>b</i>	<i>edit bsd disklabel</i>
<i>c</i>	<i>toggle the dos compatibility flag</i>
<i>d</i>	<i>delete a partition</i>
<i>l</i>	<i>list known partition types</i>
<i>m</i>	<i>print this menu</i>
<i>n</i>	<i>add a new partition</i>
<i>o</i>	<i>reate a new empty DOS partition table</i>
<i>p</i>	<i>print the partition table</i>
<i>q</i>	<i>quit without saving changes</i>
<i>s</i>	<i>create a new empty Sun disklabel</i>
<i>t</i>	<i>change a partition's system id</i>
<i>u</i>	<i>change display/entry units</i>
<i>v</i>	<i>verify the partition table</i>
<i>w</i>	<i>write table to disk and exit</i>
<i>x</i>	<i>extra functionality (experts only)</i>

Command (m for help):

Para criarmos uma nova partição devemos antes ver se temos espaço disponível para isso, ou seja, precisamos imprimir a tabela de partições utilizando a letra p. Se houver espaço disponível para a criação de uma nova partição basta pressionar a letra n e informar o tipo da partição (primária ou estendida) e seu tamanho.

5.1.2. Particionamento com CFDISK

A ferramenta cfdisk não é tão completa quanto o fdisk mas é um pouco mais "user friendly". Para acessá-la basta executar o comando:



```
# cfdisk /dev/hda
```

Uma vez executado esse comando, a tela do cfdisk se abrirá como mostrado na figura:

```

                                cfdisk 2.11n
                                Disk Drive: /dev/hda
                                Size: 10205282304 bytes
                                Heads: 255   Sectors per Track: 63   Cylinders: 1240

```

Name	Flags	Part Type	FS Type	[Label]	Size (MB)
hda1	Boot	Primary	Linux ext3		16.46
hda2		Primary	Linux ext3		501.75
hda3		Primary	Linux swap		254.99
hda5		Logical	Linux ext3		4499.23
hda6		Logical	Linux ext3		296.12
hda7		Logical	Linux ext3		197.41
hda8		Logical	Linux ext3		3997.49
		Logical	Free Space		435.94

```

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ]   [ Type ]   [ Units ] [ Write ]

Toggle bootable flag of the current partition

```

Ilustração 5: cfdisk

A utilização do cfdisk é bastante intuitiva, utilizando as setas para cima e para baixo você navega pela listagem das partições e, utilizando as setas para a esquerda e direita, você navega pelo menu na parte inferior da tela.

Para criar uma nova partição basta selecionar na listagem de partições a linha que contém espaço livre e entrar na opção **New** no menu inferior. Se ainda for possível criar partições primárias surgirá a pergunta pelo tipo da partição, caso contrário, surgirá a pergunta para especificar quanto espaço deve ser destinado para essa partição.

Após realizar todas as alterações, escolha, no menu inferior, a opção Write. Uma pergunta pedindo que você confirme as alterações irá aparecer. Sua resposta deve ser ``sim" ou ``não" com todas as três letras!! Afinal, você é o root e sabe o que está fazendo! :)

Pronto, criadas as partições precisamos aplicar um filesystem.

5.2. Aplicando um Filesystem

Para que possamos gravar informações de forma estruturada na partição que acabamos de criar precisamos aplicar um filesystem a ela. Sim, aplicar um filesystem, NÃO formatá-la!!!

Formatar é o processo de preparar a mídia magnética, como discos rígidos e disquetes, para receber informação. Esse tipo de preparo é de baixo nível e consiste em ``desenhar" as trilhas e setores na mídia em questão. Aplicar o filesystem significa criar uma estrutura lógica acima dessas trilhas e setores que permita organizar seus arquivos em uma estrutura de diretórios e subdiretórios.

Vamos conhecer alguns tipos de FileSystem

- ext2** - Um dos primeiros filesystems do linux;
- ext3 - Evolução** do ext2, mas com a técnica de Journal
- reiserfs** - Ótimo sistema de arquivos para arquivos menores que 4MB
- xfs** - Usado geralmente em banco de dados, tem suas vantagens com objetos muito grandes.



*As ferramentas de manutenção do xfs conhecidas como xfs-tools, podem de dar alguns pontos positivos na prova. O pacote referente no Debian se chama **xfsprogs***

Para criar um filesystem em uma partição deve escolher o tipo de filesystem e utilizar o comando mkfs cuja forma de utilização básica é a seguinte:



```
# mkfs -t tipo_do_FS <dispositivo>
```



Leitura sugerida: man mkfs

Sendo que o filesystem que você pode escolher para criar no device deve ser suportado pelo kernel e deve ter seu software instalado. Para consultar quais filesystem estão com suporte no kernel basta consultar o arquivo /proc/filesystems.

Dessa forma, podemos exemplificar a criação de um filesystem em um dispositivo utilizando o seguinte comando:



```
# mkfs -t ext3 /dev/hdb1
```

Aplicado o filesystem, só falta criar o ponto de montagem e montar!



Ao contrário do que pensam os file systems não mordem, e podem ser grandes aliados na prova, principalmente no termo migração de file system. Lembre-se da migração mais comum de filesystems ext2 para ext3.

5.3. Arquivos de Informações de Filesystems

Na seção `sec:mount` você aprendeu a montar um dispositivo de forma completa e manual, entretanto, há um arquivo que facilita a nossa vida, o `/etc/fstab`. Nele devem estar as informações a respeito da montagem de todos os filesystems do sistema, veja um exemplo a seguir:



```
<file system> <mount point> <type> <options> <dump> <pass>
proc                /proc proc defaults 0          0
/dev/hda1           /boot  ext3 defaults 0          1
/dev/hda2           /      ext3 defaults,errors=remount-ro 0 2
/dev/hda3           none   swapsw          0          2
/dev/hda5 /usr ext3 defaults 0          2
/dev/hda6           /var   ext3 defaults 0 2
/dev/hda7           /tmp   ext3 defaults 0          2
/dev/hda8           /home  ext3 defaults 0          0
UUID=be35a709-c787-4198-a903-d5fdc80ab2f8 /media/chas ext3
relatime,errors=remount-ro 0 1
```

As informações que devem ir nesse arquivo, de acordo com o número da coluna são:

- **localização do filesystem**, em geral o device ou endereço de rede;
- **ponto de montagem**;
- **tipo do filesystem**, ext3, reiserfs, xfs, etc;
- **opções de montagem** (defaults = rw, suid, dev, exec, auto, nouser e async).

Ver `man mount`;

- aceita os valores 0 ou 1 e informa que, havendo um sistema de backup (**dump**) configurado, deverá ser feito o seu backup;

- aceita os valores de 0 a 2 e informa que deverá ser realizada a checagem (**pass**) de integridade do sistema de arquivos. O valor zero desativa a funcionalidade, o valor 1 deve ser especificado apenas para o / e o valor 2 deve ser especificado para quaisquer outros sistemas de arquivos.

Sendo assim, o `fstab` armazena as informações dos dispositivos comumente acessados, como as partições do sistema, discos removíveis e alguns dispositivos USB, entretanto, não mostra informação alguma a respeito de quais dispositivos estão montados neste exato momento.



Essa informação pode ser obtida acessando o arquivo `/etc/mtab` ou `/proc/mounts`; ambos os arquivos são uma tabela atualizada em tempo real e que mostra quais dispositivos estão montados e com quais parâmetros.

5.4. Configurações de Teclado no Console

Imagine que você instalou uma máquina na sua casa e o seu teclado é um teclado padrão brasileiro -- ABNT2 -- e você chegou ao cliente e ele só possui teclados com layout americano; como resolver o problema?

É possível utilizar o comando `loadkeys` para alterar o layout de teclado durante a sessão mas, essa alteração será temporária. Para trocar definitivamente o padrão de layout de teclado da máquina, o arquivo em `/etc/console/boottime.kmap.gz` deve ser alterado utilizando o comando `kbd-config`.

Além da configuração apropriada de layout de teclado, pode ser interessante configurar o mouse em modo texto a fim de facilitar o trabalho. O programa chamado `gpm` é o que dá o suporte ao mouse em terminais texto.

5.5. Prática Dirigida 1

1) Altere o layout de teclado para utilizar o padrão americano:



```
# loadkeys -d us
```

Tente utilizar a tecla ``ç''

2) Volte o layout de teclado para o padrão br-abnt2:



```
# loadkeys -d br-abnt2
```

3) Altere o layout de teclado padrão do sistema. Você tem duas opções:



```
# kbd-config  
# dpkg-reconfigure console-data
```

5.6. Para aprofundar o assunto

Parâmetros de montagem:



```
# info mount
```

O arquivo /etc/fstab:



```
# info fstab
```

5.7. Prática Dirigida 2

1) Determine qual o dispositivo associado ao "hd", tanto o CD-ROM, quanto o Disco Rígido:



```
# dmesg |grep hd
```

2) Coloque um CD no drive e torne o conteúdo acessível no diretório /media/cdrom:



```
# mount -t iso9660 <dispositivo> /media/cdrom
```

3) Verifique que o dispositivo foi montado:



```
# mount  
# df -h  
# cat /etc/mtab  
# cat /proc/mounts
```

4) Entre no diretório e explore o conteúdo do CD:



```
# cd /media/cdrom ; ls
```

5) Desmonte o CD:



```
# umount /media/cdrom
```

Deu erro?? Por quê??

6) Saia do diretório /media/cdrom:



```
# cd
```

7) Tente desmontá-lo novamente:



```
# umount /media/cdrom
```

Agora sim!

8) Monte o CD novamente, entre no diretório do ponto de montagem e, de dentro dele, abra uma nova shell:



```
# mount -t iso9660 <dispositivo> /media/cdrom  
# cd /media/cdrom  
# bash
```

9) Saia do diretório e desmonte o CD:



```
# cd  
# umount /media/cdrom
```

Funcionou?? E agora??

Agora que já sabemos montar dispositivos, vamos aprender a criar partições.

10) Utilizando o particionador 'cfdisk' crie uma nova partição de 'swap' e uma nova partição linux:

As novas partições devem ter os seguintes tamanhos:

- **/dev/hda9** - partição tipo swap com 256MB;
- **/dev/hda10** - nova partição com 1000;
- espaço vazio não particione.



```
# cfdisk /dev/hda
```

Após criar as novas partições será necessário rebotar a máquina para que a nova tabela de partições seja relida. No nosso caso, a versão do Kernel é a 2.6.18, então estamos utilizando udev, mesmo sendo udev, temos que reiniciar.

Com a tabela de partições atualizadas podemos aplicar os filesystems e paginar a nova partição de swap.

1) Prepare a partição de swap:



```
# mkswap <dispositivo>
```

2) Ative essa nova partição de swap:



```
# swapon <dispositivo>
```

O filesystem que desejamos aplicar a uma das novas partições é o ext3. Para que possamos realizar essa tarefa devemos determinar se o nosso kernel suporta este filesystem e se o software necessário está instalado.

3) Determine se o ext3 pode ser utilizado, ou seja, tem suporte no kernel:



```
# ls /proc/filesystems
```

4) Instalados os softwares podemos aplicar o ext3 à nova partição:



```
# mkfs -t ext3 <dispositivo>
```

Uma vez que o filesystem foi aplicado à partição, vamos torná-lo acessível por meio do diretório /backup.

5) Crie o ponto de montagem /backup:



```
# mkdir /backup
```

6) Teste a montagem do novo filesystem:



```
# mount -t ext3 <dispositivo> /backup
```

7) Verifique se a partição foi montada e se o swap está em uso:



```
# mount  
# df -h  
# cat /etc/mtab  
# cat /proc/mounts  
# cat /proc/swaps
```

8) Crie um arquivo dentro do /backup:



```
# touch /backup/README  
# echo "Partição de Backup" >> /backup/README
```

9) Coloque as entradas no fstab para que o novo swap e a partição de backup sejam montadas na hora do boot:



```
<dispositivo> none swapsw 0 0  
<dispositivo> /backup ext3 defaults 0 0
```



O arquivo `/etc/fstab` é criado automaticamente quando o Linux é instalado. Assim gerando uma lista de partições padrões a serem carregadas.