

## GERENCIAMENTO DE PARTIÇÕES

Podemos criar partições utilizando o comando *fdisk* seguido do dispositivo que queremos particionar:

```
fdisk /dev/sda
```

O *fdisk* é um comando interativo e seu menu de opções pode ser acessado digitando a letra *m*. Após criada a partição, ela receberá um número de acordo com a ordem de criação. */dev/sda1* é a primeira partição do dispositivo */dev/sda*, */dev/sda2* é a segunda e assim por diante.

Após criada a partição devemos formatar usando o sistema de arquivos desejado. Para isso, podemos usar o comando *mkfs.<fs>*, onde *<fs>* deve ser trocado pelo sistema de arquivos correspondente. Por exemplo, para formatar a partição */dev/sda8* com o sistema de arquivos *ext4*, usamos:

```
mkfs.ext4 /dev/sda8
```

Feito isso, podemos montar a partição em qualquer diretório desejado. Exemplo:

```
mkdir /dados1  
mount /dev/sda8 /dados1
```

O comando *mount* faz a montagem da partição em tempo de execução. Para tornar a montagem persistente podemos criar um script com o comando *mount* desejado e adicionar esse script na inicialização. Mas esse método não é indicado, uma vez que podemos usar o *fstab* para concentrar todas as tarefas de montagem do sistema. O *fstab* será explicado a seguir.

## MONTAGEM DE SISTEMAS DE ARQUIVOS - FSTAB

O arquivo */etc/fstab* mantém as informações das partições que devem ser montadas no Linux. Essa montagem ocorre durante o boot (exceto se a opção *noauto* estiver explícita).

Abaixo, um exemplo do arquivo:

# device name	mount point	fs-type	options	dump-freq	pass-num
/dev/sda1/	/	ext4	defaults	1	1
/dev/hda6	swap	swap	defaults	0	0

As colunas possuem os seguintes itens:

1. O path, label ou UUID do dispositivo que vai ser montado.
2. O ponto de montagem (diretório) onde o dispositivo será montado.
3. O tipo do sistema de arquivos ou o algoritmo usado para interpretar o sistema de arquivos.
4. Opções, incluindo se o sistema de arquivos será montado durante o boot.
5. Frequência com que será executado o arquivamento para a partição (dump). “0” desabilita o dump.
6. A ordem que o *fsck* irá checar as partições. A partição “/” deve ser 1 e as demais “2”. “0” desabilita a checagem.

## MEMÓRIA VIRTUAL SWAP

A memória swap é um espaço em disco usado pelo SO para fazer uma extensão ad memória RAM, quanto esta está escassa. Por ter um desempenho bastante inferior ao desempenho da memória RAM, o uso da swap acarreta em lentidões no sistema.

No Linux é possível utilizar swap em partições dedicadas ou em arquivos de imagem.

Em partição:

- Deve existir uma partição disponível, com o tamanho desejado.
- Formatamos essa partição:

```
mkswap /dev/sda2
```

- Ativamos a swap:

```
swapon /dev/sda2
```

- Tornamos persistente adicionando a seguinte linha no arquivo */etc/fstab*:

```
/dev/sda2  swap  swap  defaults  0 0
```

Em arquivo:

- Criamos um arquivo:

```
dd if=/dev/zero of=/var/arqswap bs=1M count=1024
```

- Formatamos o arquivo:

```
mkswap /var/arqswap
```

- Ativamos a swap:

```
swapon /var/arqswap
```

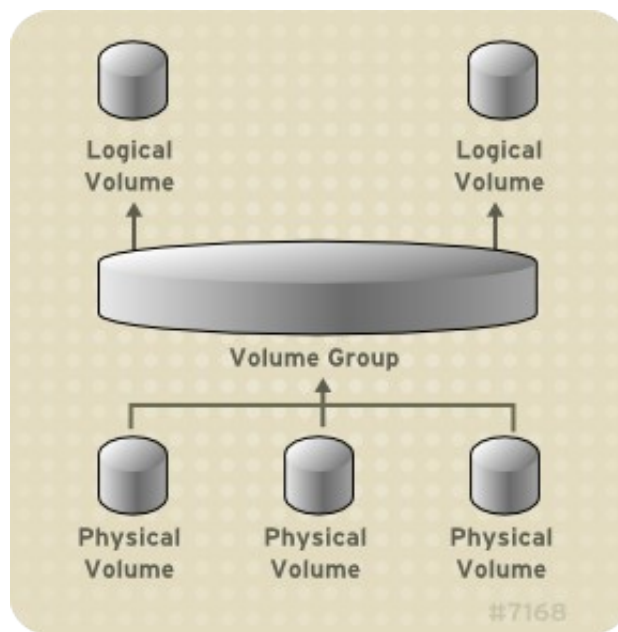
- Tornamos persistente adicionando a seguinte linha no arquivo `/etc/fstab`:

```
/var/arqswap  swap  swap  defaults  0 0
```

Verificando:

```
free -m  
cat /proc/swaps
```

### LVM



O LVM cria uma camada de abstração sobre o hardware de armazenamento. Assim, você pode criar volumes de armazenamento lógicos, oferecendo maior flexibilidade. O LVM elimina a restrição aos tamanhos de discos físicos.

Dentre as vantagens em relação ao armazenamento físico, podemos destacar:

- Capacidade flexível.
- Nomeação conveniente de dispositivos.
- Capacidades de armazenamento redimensionáveis.
- Uso simultâneo de discos.
- Snapshots e espelhamento.

Para criar uma estrutura LVM, devemos primeiro registrar os dispositivos físicos:

```
pvccreate /dev/sda9 /dev/sda10 /dev/sda11  
pvs
```

Em seguida, criamos nosso Volume Group usando os dispositivos necessários:

```
vgcreate vg01 /dev/sda9 /dev/sda10  
vgs
```

Por último, criamos os Logical Volumes de acordo com nossas necessidades, especificando em que Volume Group eles serão criados e o tamanho que ficará disponível:

```
lvcreate vg01 -n lv01 -L 200M  
lvs
```

Depois de criado o Logical Volume, podemos formatar e montar o dispositivo resultante:

```
mkfs.ext4 /dev/vg01/lv01  
mkdir /dados2  
mount /dev/vg01/lv01 /dados2
```

Para expandir um volume existente, primeiro temos que expandir o Logical Volume e depois informar isso para o sistema de arquivos. Nesse caso não é preciso informar o novo tamanho para o *resize2fs*, pois ele usará todo o espaço disponível. Podemos fazer essa operação com a partição montada.

```
lvextend -L +100M /dev/vg01/lv01  
resize2fs -p /dev/vg01/lv01
```

Para reduzir, teremos que desmontar a partição . Depois de desmontada, é necessário rodar o *e2fsck* para garantir que o sistema de arquivos não será corrompido:

```
umount /dev/vg01/lv01  
e2fsck -f /dev/vg01/lv01
```

Agora, a redução propriamente dita, seguindo o caminho inverso da expansão. Primeiro, reduzimos o sistema de arquivos para o novo tamanho e depois reduzimos o Logical Volume:

```
resize2fs /dev/vg01/lv01 128M  
lvreduce -L 128M /dev/vg01/lv01
```

Feito isso, podemos montar novamente o volume:

```
mount /dev/vg01/lv01 /dados2
```

Lembrando que a montagem com o comando *mount* não é persistente. Assim, use o arquivo */etc/fstab* para fazer a montagem no processo de boot.

### **RAID VIA SOFTWARE**

O RAID via software permite ao Sysadmin se beneficiar do uso de arrays de RAID sem possuir uma controladora com esse recurso, como aumento da velocidade de leitura e gravação, uso de dispositivos reservas e redundância de discos. A desvantagem desse método é que as informações sobre os arrays ficam no sistema operacional. Além disso, o processamento relacionado aos arrays será realizado no processador do computador, o que pode vir a causar perdas de desempenho em sistemas sobrecarregados.

Para criar um array de RAID, usaremos o comando *mdadm*. No exemplo abaixo, criaremos um array RAID5 com 3 discos e mais 1 disco de spare:

```
mdadm -C /dev/md0 -n 3 -l 5 -x 1 /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

O nome do dispositivo resultante é */dev/md0*. O *-n* indica o número de dispositivos do array, o *-l* indica o nível de RAID e o *-x* indica quantos dispositivos de spare teremos. Por último, indicamos o path das 4 partições envolvidas. Podemos verificar o estado do nosso array com os comandos abaixo:

```
mdadm --detail /dev/md0  
cat /proc/mdstat
```

Para efeito de testes, podemos provocar a falha em uma das partições. Veja o exemplo:

```
mdadm /dev/md0 -f /dev/sda1
```

Quando um dispositivo falhar, devemos removê-lo e adicionar um substituto:

```
mdadm /dev/md0 -r /dev/sda1
```

```
mdadm /dev/md0 -a /dev/sde1
```

No exemplo aqui mostrado, a falha de um dos dispositivos resultará no uso do dispositivo spare. Caso não haja nenhum dispositivo de spare, o array RAID5 continuará funcionando devido a sua natureza. Nesse ponto, a falha de um segundo dispositivo seria fatal para o array.