

# RAID

## Tutorial RAID por software no Linux

Autor: Marcio Katan

Técnico de Suporte Linux e Instrutor Certificado Conectiva Mandriva

Autor do livro: Linux no Computador Pessoal com Conectiva 10

*marcio\_katan@yahoo.com.br*

Rio de Janeiro – RJ

### Redundant Array of Inexpensive Disks – Arranjo (Matriz) Redundante de Discos Independentes

A tecnologia RAID é utilizada para combinar dois ou mais (“vários”) discos (HDs) ou partições em um arranjo formando uma única unidade lógica (matriz) para armazenamento de dados. A idéia básica do RAID consiste em dividir a informação em unidades e, em caso de falha de um destes, uma outra unidade assume a que falhou.

Quando uma informação é endereçada ao arranjo RAID, ela será segmentada e armazenada de forma distribuída nas unidades que formam o arranjo. Esta informação segmentada é dividida de acordo com o que chamamos de “**stripe**” (pedaço). A junção (soma) dos stripes dos dispositivos formam o **chunk** do dispositivo RAID. O tamanho do **chunk** é medido em KB (kiloBytes), variando de 4 (para 4 KB) a 4096 (para 4MB). Se uma informação **A** tem 16KB de tamanho, e o chunk do arranjo for de 18KB, cada segmento (stripe) da informação será segmentado em 8KB cada e armazenado nos dispositivos. Quando não é definido o tamanho do chunk, o sistema assume como padrão o valor de 64KB. O chunk para o dispositivo RAID é o mesmo que o bloco para o sistema de arquivo.

O dispositivo virtual (matriz/arranjo) criado para gerenciar o RAID chama-se MD (multiple device – dispositivo múltiplo).

A tecnologia RAID foi desenvolvida na universidade de Berkeley – Califórnia – a mais de 15 anos.

Existem dois tipos de RAID:

- **Por HARDWARE**

Este tipo de RAID é implementado, principalmente nas controladoras SCSI. Praticamente, todas as controladoras SCSI possui RAID por hardware. Algumas placas-mães (como ABIT, SOYO e ASUS) trazem consigo, suporte a RAID para unidades IDE. As controladoras mais utilizadas nestas placas, são a HPT e PROMISSE. Todas as controladoras SATA trazem suporte a RAID.

- **Por SOFTWARE**

Neste tipo de RAID, o arranjo é controlado pelo kernel do sistema operacional. O kernel do Linux suporta RAID pelos softwares (ferramentas) *raidtools* e/ou *mdadm*. As distribuições de versões mais antigas, utilizavam o *raidtools*. Hoje, praticamente todas as distribuições utilizam o *mdadm* (multiple devices admin). E é esta ferramenta que será abordada neste tutorial.

As distribuições utilizadas para os testes foram o Mandriva 2006 Power Pack e Debian Sarge 3.1 r0.

O RAID é dividido em níveis, variando de RAID Linear, 0, 1, 2, 3, 4, 5 a suas formas combinadas (híbridas): 0+1, 1+0, 5+0. Neste tutorial, veremos os conceitos dos RAIDs Linear, 0, 1, 2, 3, 4 e 5.

Para a implementação do RAID, são necessários no mínimo dois dispositivos. Alguns níveis necessitam três, quatros ou até mais dispositivos!

**OBS.:** Embora seja explicado neste tutorial os níveis de RAID 2 e 3, estes não são suportados pelo Linux

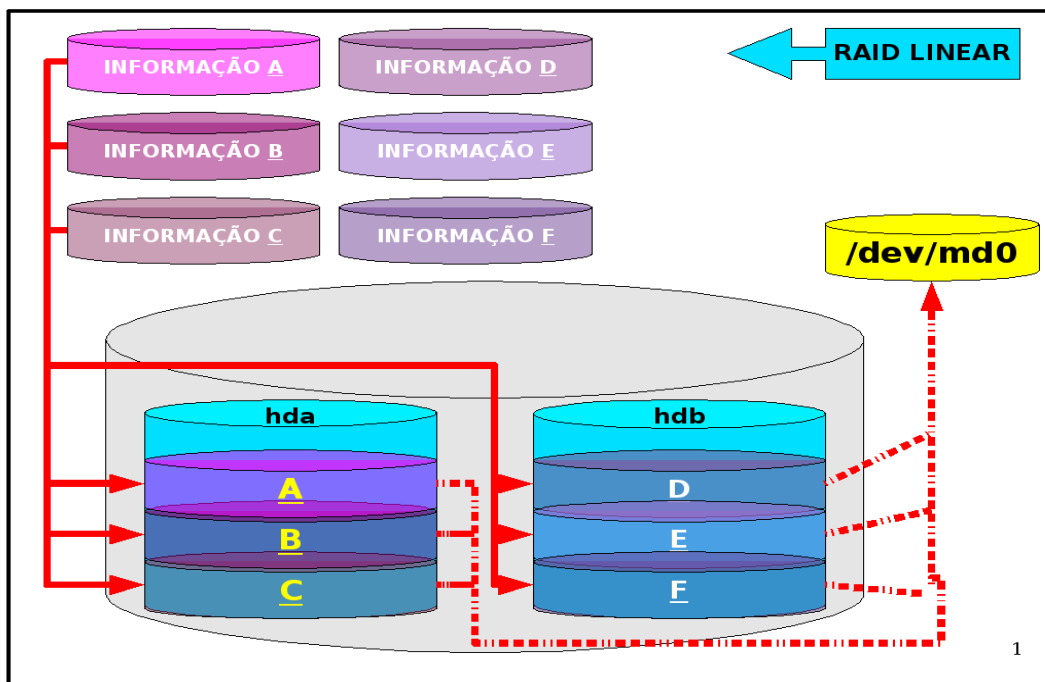
Vejamos uma breve explicação sobre os níveis de RAID:

- **RAID Linear**

Foi a primeira “tentativa” de RAID. RAID Linear nada mais é do que a concatenação (junção) de discos ou partições para formar um único arranjo virtual. Com o advento do LVM (Logical Volume Manager – Gerenciador de Volume Lógico) o RAID Linear ficou obsoleto.

Neste nível, a informação é gravada no primeiro dispositivo até completá-lo e, ao ocupar o primeiro dispositivo, segue seu armazenamento na segunda unidade e assim sucessivamente.

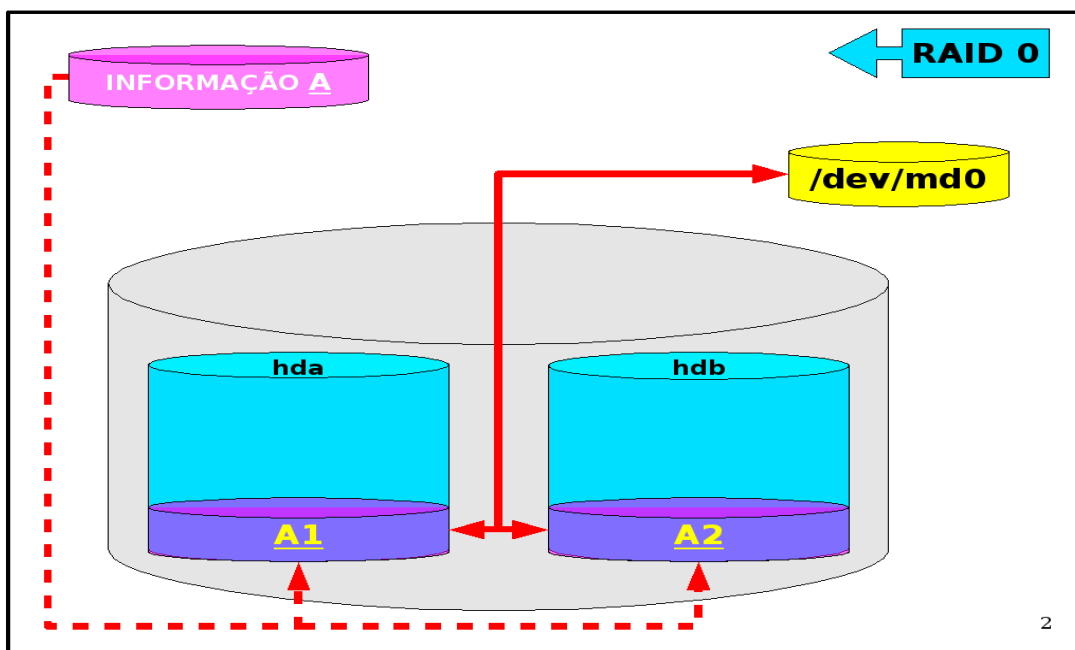
Cálculo do RAID Linear: HD 20GB + HD 20GB = /dev/md0 40GB



- **RAID 0 – DATA STRIPPING**

No nível RAID 0, a informação é segmentada e, cada segmento, armazenado em cada unidade do arranjo. Neste nível, não há redundância, pois há somente uma divisão (stripping) da informação. A única vantagem do RAID 0 é o ganho de velocidade no acesso a informação.

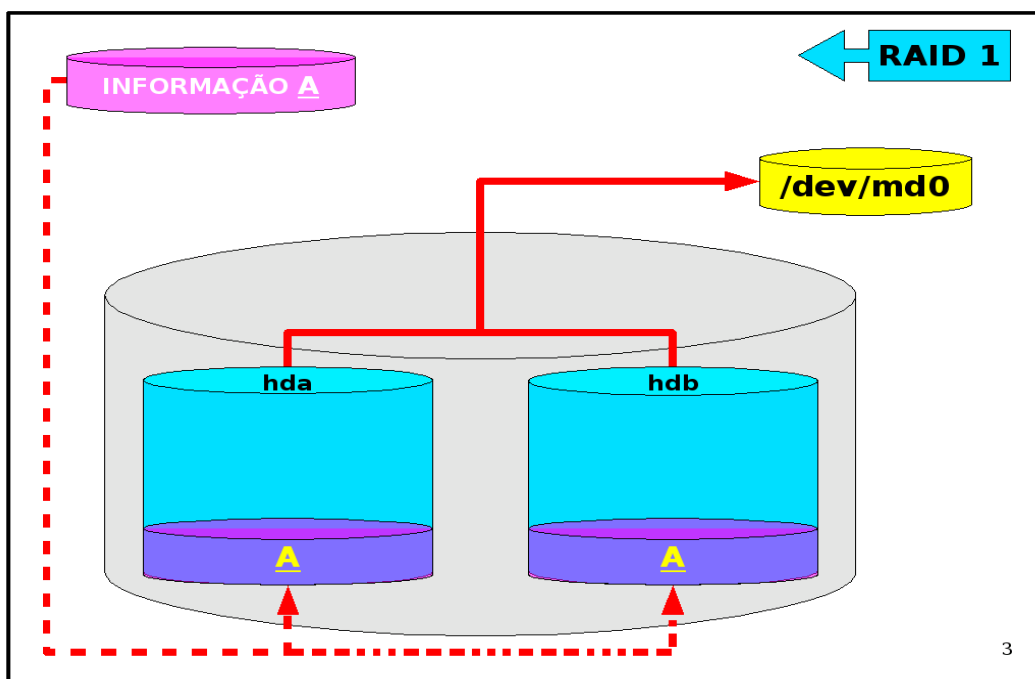
HD 20GB + HD 20GB = /dev/md0 40GB



- **RAID 1 – DATA MIRRORING**

No RAID nível 1, a informação é gravada igualmente em todas as unidades (mirror – espelho). À partir deste nível, há redundância de dados, pois com a duplicidade da informação, quando uma unidade falha, o seu espelho (mirror) assume.

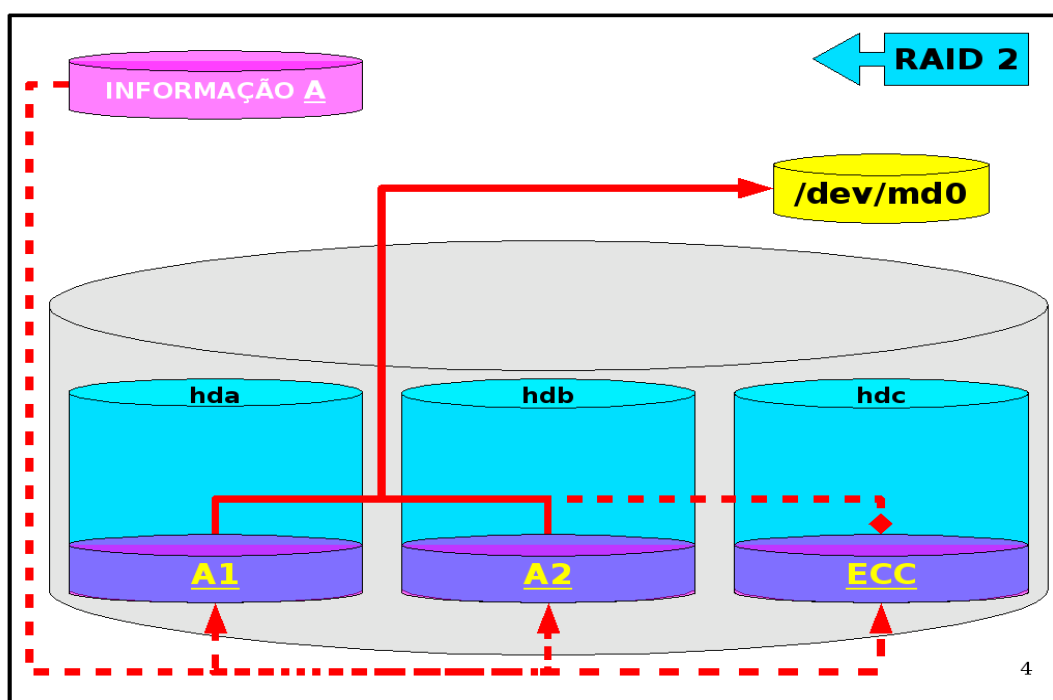
HD 20GB + HD 20GB = /dev/md0 20GB



- **RAID 2 – DATA STRIPPING WITH ECC**

O nível RAID 2 está obsoleto. Este nível consiste em segmentar a informação pelos dispositivos e, ao gravar a informação, é gravado em uma unidade extra (dedicada) uma informação de ECC (Error Correcting Code – Código de Correção de Erro). Como todos os HDs atuais possuem a tecnologia de ECC para gravação de dados, este nível ficou obsoleto.

HD 20GB + HD 20GB + HD 20 GB (ECC) = /dev/md0 40GB

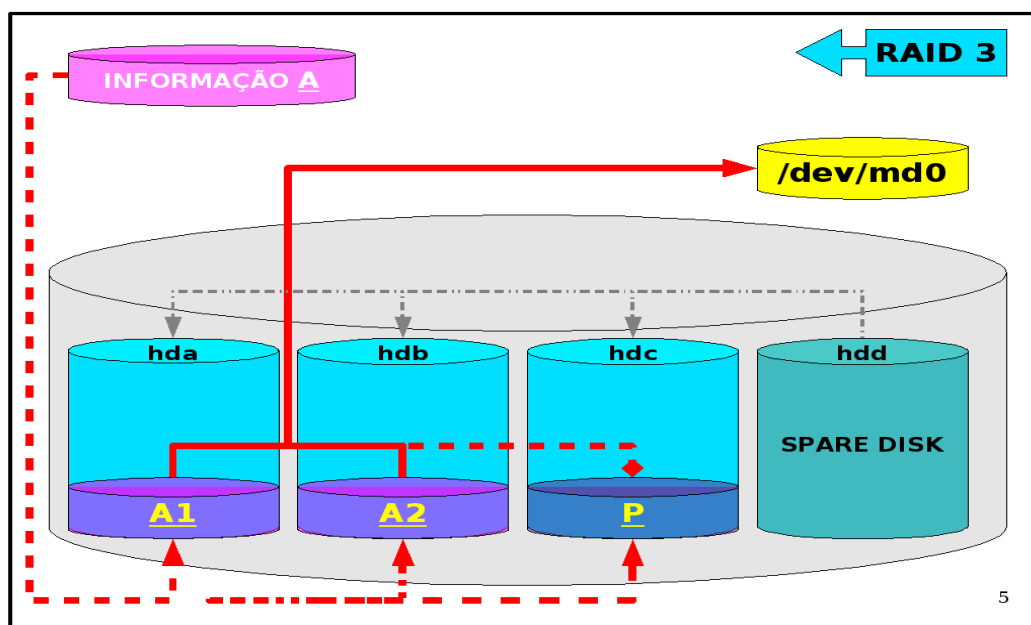


- **RAID 3 – DATA STRIPPING WITH PARITY DISK**

No RAID nível 3, quando a informação é segmentada pelos dispositivos, esta segmentação é feita por grupos de bits e é gerado uma informação de paridade em um disco dedicado para reconstrução da informação quando um dispositivo vier falhar no arranjo. A reconstrução é feita em um outro disco chamado de *spare disk* (disco estepe – reserva). Embora o bit de paridade seja utilizado para reconstruir a informação, serão necessários quatro discos (unidades) para uma “perfeita” implementação do RAID nível 3. Dois discos para a implementação da segmentação, um para a paridade e um para o spare.

Pelo fato da implementação do RAID 3 gerar um segmentação muito elevada da informação (pedaços – stride – muito pequenos), este nível gera muito acesso de E/S, tornando-o muito lento em determinadas situações. Por este motivo, o RAID 4 o substituiu.

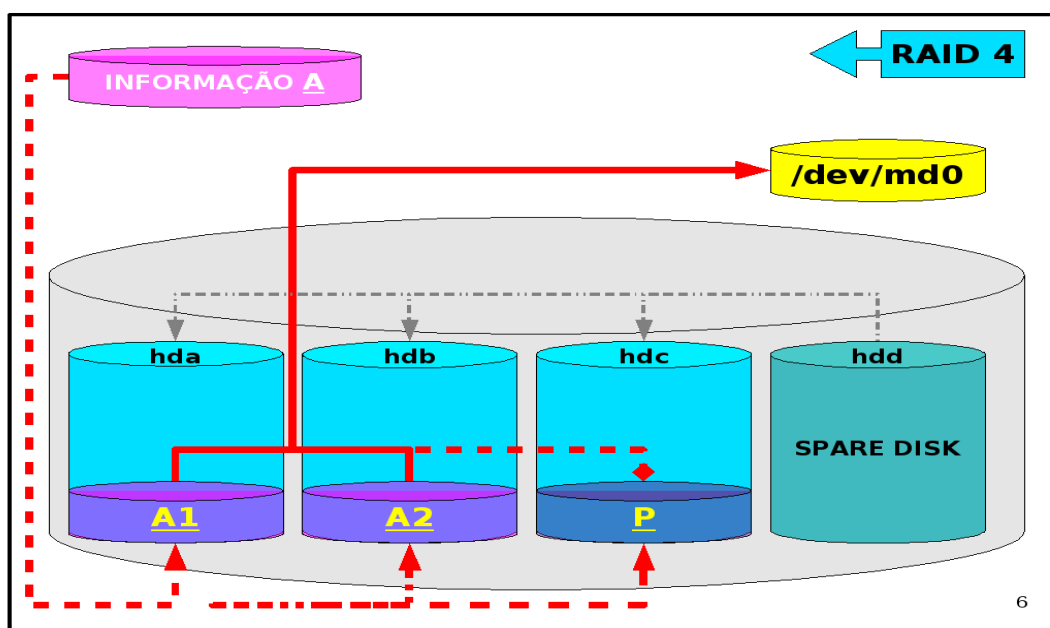
**HD 20GB + HD 20GB + HD 20GB (PARIDADE) + HD 20 GB (SPARE) = /dev/md0 40GB**



- **RAID 4**

O nível de RAID 4 é semelhante ao nível 3. A diferença está na segmentação dos dados (maior no nível 4) e a construção da informação pela paridade, que é feita em tempo real quando um dispositivo falha.

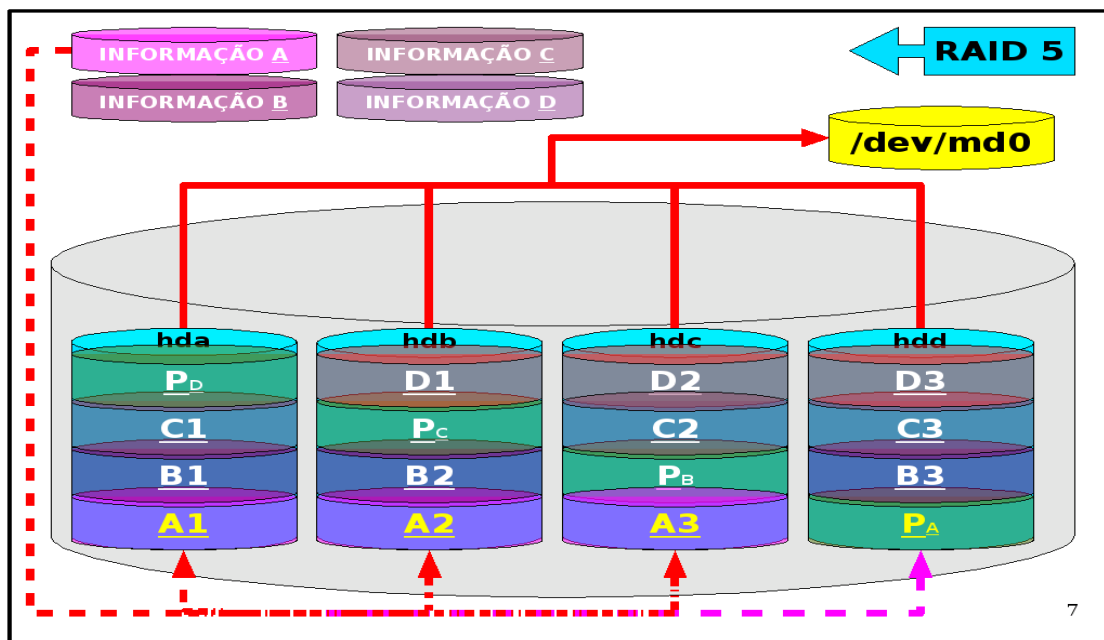
**HD 20GB + HD 20GB + HD 20GB(P) + HD 20GB(SPARE) = /dev/md0 40GB**



- **RAID 5**

No RAID 5, surge uma nova implementação da segmentação da informação e do uso (armazenamento) da paridade. Neste nível, a paridade não é mais armazenada em um único disco dedicado. Um algoritmo é utilizado para segmentar a informação e calcular a paridade. Se um arranjo RAID tem 5 dispositivos, o primeiro bloco da informação será segmentado pelos quatro primeiros dispositivos e a paridade armazenada no quinto. No segundo bloco de dados, este será segmentado e armazenado até o terceiro bloco, pois o quarto bloco, que seria armazenado no quarto dispositivo será utilizado para guardar o segmento de paridade e o quarto segmento de dado será gravado no quinto dispositivo. O terceiro bloco de dados segue o mesmo algoritmo (raciocínio). Será segmentado e sua paridade é gravado no dispositivo anterior, e assim sucessivamente (veja desenho abaixo.)

O algoritmo utilizado para a paridade utiliza cerca de 30% de espaço em disco para armazenar a paridade.



## MÃOS À OBRA

Antes de criar o dispositivo raid, deve-se “sinalizar” (marcar) as partições com sendo de uso para o raid.

**OBS.:** Em caso de utilização de HDs inteiros para compor os dispositivos do arranjo, deve-se criar uma única partição no HD.

Com a ferramenta fdisk marque os dispositivos como sendo do tipo **fd Detecção automática de RAID** – o código é em hexadecimal (0x0fd). Para mais detalhe de como utilizar o fdisk, veja sua página man.

### Criando o arranjo:

Ferramenta *mdadm*

```
mdadm [-C | --create] [-v | --verbose] /dev/mdX [-l N | --level=N] [-c N | --chunk=N]
[-n N D1 D2 ... Dn | --raid-devices=N D1 D2 ... Dn]
[-p <algorithm>] [-x N Dx1 Dx2 ... Dxn | --spare-devices=N Dx1 Dx2 ... Dxn]
```

Opções:

**-C ou --create**

Esta opção cria o dispositivo. O primeiro dispositivo raid será sempre **/dev/md0**. Qualquer uma das opções podem ser utilizadas (**-C** ou **--create**)

Ex.: mdadm -C  
mdadm --create

**-v ou --verbose**

Esta opção mostra em detalhes a criação do arranjo

Ex.: mdadm -v  
mdadm --verbose

**/dev/mdX**

O arranjo em */dev*. O **X** do **mdX** começa em 0 (zero) e segue de acordo com ordem de criação.

Ex.: mdadm -Cv /dev/md0

**-l N ou --level=N**

Nível do raid a ser implementado no arranjo. **N** é o número do nível - 0, 1, 4, 5 - ou **linear** para RAID Linear). Pode-se utilizar a forma **raid0**, **raid1**, **raid4**, **raid5**.

Ex.: -l 5  
--level=raid5

**-c N ou --chunk=N**

Tamanho do chunk utilizado no arranjo. Se omitido, o sistema assume o valor de 64 (KB)

Ex.: -c 32  
--chunk=32

**-n N <device\_1> <device\_2> <device\_N> ou --raid-devices=N <device\_1> <device\_2> <device\_N>**

Quantidade e lista dos dispositivos que irão compor o arranjo raid. **N** para o total de dispositivos que irão compor o arranjo.

Ex.: -n 3 /dev/hdb1 /dev/hdc1 /dev/hdd1  
--raid-devices=3 /dev/hdb1 /dev/hdc1 /dev/hdd1

**-p <algorithm> ou parity=<algorithm>**

Algoritmo utilizado para cálculo da paridade. Somente utilizado no RAID nível 5.

Algoritmos possíveis: **left-asymmetric (la)**, **right-asymmetric (ra)**, **left-symmetric (ls)** ou **right-symmetric (rs)**. Se omitido, o sistema assume **left-symmetric** como padrão. E este é o algoritmo de melhor performance.

Ex.: -p left-symmetric  
--parity=left-symmetric

**-x N <device\_1> <device\_2> <device\_N> ou --spare-disks=N <device\_1> <device\_2> <device\_N>**

Quantidade e lista dos dispositivos que serão utilizados como dispositivo reserva (spare). Caso se esteja configurando um arranjo RAID 4, de acordo com o man do mdadm, o disco spare será utilizado como disco de armazenamento da paridade. Esta informação não está plenamente confirmada!

Ex.: -x 2 /dev/hde /dev/hdf  
--spare-disks=2 /dev/hde /dev/hdf

## Criando o arranjo RAID

### RAID 0

```
mdadm --create --verbose /dev/md0 --level=0 --chunk=4 --raid-devices=2 /dev/hda2 /dev/hdb2
ou
mdadm -C -v /dev/md0 -l 0 -c 4 -n 2 /dev/hda2 /dev/hdb2
```

Por questões óbvias, o RAID 0 não tem disco **spare**.

### RAID 1

```
mdadm --create --verbose /dev/md0 --level=raid1 --chunk=16 --raid-devices=2 /dev/hdb /dev/hdd
ou
mdadm -C -v /dev/md0 -l 1 -c 4 -n 2 /dev/hda1 /dev/hdb1
```

\* se tiver disco **spare**, acrescente: `--spare-devices=1` ou `-x 1 /dev/hdc2`

### RAID 5

```
mdadm -C -v /dev/md0 -l 5 -c 32 -p ls -n 4 /dev/hda1 /dev/hdb1 /dev/hdc1 /dev/hdd1
ou
mdadm --create --verbose /dev/md0 --level=raid5 --chunk=32 --parity=left-symmetric -raid-devices=4
/dev/hda1 /dev/hdb1 /dev/hdc1 /dev/hdd1
```

\* se tiver disco **spare**, acrescente: `--spare-devices=1` ou `-x 1 /dev/hdc2`

Ao finalizar a criação do arranjo, deve-se editar o arquivo `/etc/mdadm.conf` (no Mandriva) (`/etc/mdadm/mdadm.conf` no Debian) para o gerenciamento do arranjo:

Opções do arquivo:

### DEVICE

Opção **DEVICE** deve conter os dispositivos que compõem o arranjo. Não se acrescenta os dispositivos **spare** aqui.

```
DEVICE <device1> <device2> <deviceN>
ou
DEVICE /dev/hd[abcd]1
```

```
Ex.:  DEVICE /dev/hda1 /dev/hb1 /dev/hdc1
      ou
      DEVICE /dev/hd[abc]1
```

### ARRAY

Opção **ARRAY** deve conter a lista dos dispositivos do arranjo e também os dispositivos **spare**.

```
ARRAY <array_raid> <level=N> devices=<device1>,<device2>,<deviceN>
```

```
Ex.:  ARRAY /dev/md0 level=5 devices=/dev/hda1,/dev/hdb1,/dev/hdc1,/dev/hde1,/dev/hdf1
```

## MAILADDR

Email do administrador para avisos em caso de algum problema com o arranjo.

**MAILADDR <email>**

Ex.: MAILADDR marcio\_katan@yahoo.com.br

## Gerenciamento:

Para visualizar os detalhes do arranjo:

**mdadm -D /dev/md0**

ou

**mdadm -detail /dev/md0**

Para interromper (Stop) o arranjo

**mdadm -S /dev/md0**

ou

**mdadm --stop /dev/md0**

Para re-inicializar (Run) o arranjo

**mdadm -R /dev/md0**

ou

**mdadm --run /dev/md0**

**ATENÇÃO:** Esta opção só irá funcionar se o arquivo mdadm.conf estiver configurado corretamente.

Para simular uma falha

**mdadm -f /dev/md0 /dev/hda2**

ou

**mdadm --manage --set-faulty /dev/md0 /dev/hda2**

Removendo dispositivo do arranjo

**mdadm /dev/md0 -r /dev/hda2**

Adicionado dispositivos ao arranjo

**mdadm /dev/md0 -a /dev/hda2**

Para remover e adicionar ao mesmo tempo um dispositivo ao arranjo:

**mdadm /dev/md0 -r /dev/hdb1 -a /dev/hdc1**

## Formatando:

Atenção especial deve ser dada ao formatar um dispositivo raid (md0).

O aplicativo mkfs sempre formata, por padrão, com blocos de 4096B (4KB). Como o arranjo raid tem um “pseudo” bloco (chunk) criado de tamanho variável, estes dois valores (do chunk e do bloco do mkfs) devemos passar um parâmetro ao mkfs para o sistema de arquivos do arranjo não dar problemas.

É a opção **-R stride=N**, onde **N** é um valor que, multiplicado pelo valor do bloco do mkfs, deve ser atingir o valor do chunk.

Por exemplo: Se o chunk do arranjo for de 32KB, o valor do stride será de 8(KB). Pois, 4 (KB do bloco do mkfs), multiplicado por 8 (KB) será igual a 32 (KB do chunk do arranjo.)

Ex.: **mkfs -b 4096 -R stride=8 /dev/md0**

## Referências:

[http://www.acnc.com/04\\_00.html](http://www.acnc.com/04_00.html)

Um ótimo artigo (em inglês) sobre os níveis de RAID. Há, inclusive, slides animados exemplificando cada nível.

<http://unthought.net/Software-RAID.HOWTO/>

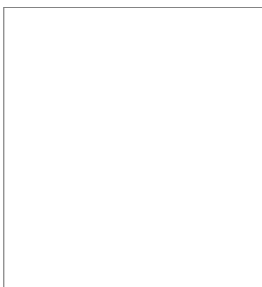
The Software RAID HOW\_TO

<http://www.conectiva.com/doc/livros/online/9.0/servidor/raid.html>

Guia on-line da Conectiva sobre RAID com a ferramenta **raidtools**.

Página man do **mdadm**

## Sobre o autor:



- Marcio Cantanhêde, conhecido como Marcio Katan, é certificado Conectiva Mandriva em Administração de Sistemas (Instrutor), Consultor e Técnico de Suporte em Linux.
- Autor do livro “Linux no Computador Pessoal com Conectiva 10” Editora Ciência Moderna

Usuário de Linux há 7 anos, a 5 aboliu o Microsoft Windows de seu computador e hoje vive feliz com o sistema GNU/Linux.

Instrutor de Redes e Linux do Senac Rio e NSI Training

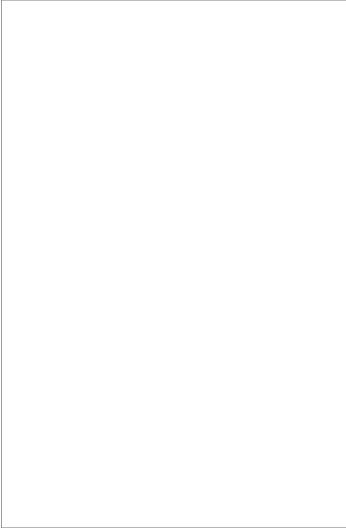
### **Contato:**

marcio\_katan@yahoo.com.br MSN: marcio\_katan@hotmail.com

Cel.: 9123-7454

Quer usar Linux em seu computador pessoal?

Compre o livro ***Linux no Computador Pessoal com Conectiva 10***



Com presença marcante nos servidores das empresas, e agora nos desktops corporativos, o sistema operacional GNU/Linux começa a travar a maior de todas as suas batalhas: a conquista do computador caseiro.

Tido como difícil de usar, este mito começa a ser quebrado com esta obra. Veremos neste livro o quão fácil é utilizar o GNU/Linux.

Tratado de forma simples e direta, o uso do sistema irá parecer brincadeira de criança. Veremos como substituir todas as funcionalidades do Windows pelo GNU/Linux. Neste existe um substitutivo para quase todos os programas que você utiliza na plataforma Microsoft.

**Sumário:**

Capítulo 1 – Iniciando o mundo GNU/Linux; Capítulo 2 – Instalando o Linux; Capítulo 3 – Conhecendo o Conectiva 10; Capítulo 4 – Configurações; Capítulo 5 – Instalando, removendo e atualizando programas; Capítulo 6 – Internet; Capítulo 7 – Impressão; Capítulo 8 – Multimídia e Entretenimento; Capítulo 9 – Programas Office; Capítulo 10 – Outros programas.

Editora: **Ciência Moderna**