



Manter a integridade dos sistemas de arquivos

Sumário

Capítulo 1

Manter a integridade dos sistemas de arquivos	3
1.1. Mãos a obra.....	4

Capítulo 2

Gerenciando	5
2.1. Objetivos.....	5
2.1. Troubleshooting.....	5

Índice de tabelas

Índice de Figuras

Capítulo 1

Manter a integridade dos sistemas de arquivos

- *Verificar a integridade dos arquivos;*
- *Monitorar espaço livre e inodes;*
- *Reparar problemas simples de arquivos.*

1.1. Mãos a obra

A administração de espaço em disco no dia a dia, é algo que o sysadmin precisa fazer, afim de não faltar espaço para a gravação de dados. No Linux é possível usar alguns comandos para realizar essa tarefa Vamos a prática:

O comando `df` é usado para exibir o espaço livre e utilizado em cada partição que estão montadas de forma local e remotas:



```
# df
```

Sist. Arq.	1K-blocos	Usad	Dispon.	Uso%	Montado em
/dev/sda2	2885812	163880	2575340	6%	/
tmpfs	258148	0	258148	0%	/lib/init/rw
udev	10240	136	10104	2%	/dev
tmpfs	258148	0	258148	0%	/dev/shm
/dev/sda1	241116	20974	207694	10%	/boot
/dev/sda3	2885812	71840	2667380	3%	/home
/dev/sda7	980308	17664	912848	2%	/tmp
/dev/sda5	3367972	1724612	1472272	54%	/usr
/dev/sda6	3367972	402092	2794792	13%	/var
/dev/sda8	3367972	75604	3121280	3%	/var/log

A saída do comando exibe informações como sistema de arquivos, tamanho da partição, espaço livre, ocupado, uso em porcentagem e o ponto de montagem.

Veja algumas opções de uso do comando `df` (disk free)

`-h` - Exibe o espaço livre e ocupado em MB, KB, GB ao invés de blocos;

`-k` - Lista em Kbytes;

`-m` - Lista em Mbytes;

`-l` - Somente lista sistema de arquivos locais.

Exemplo de uso do `df` com a flag `-h`



```
# df -h
```

```
Sist. Arq.      Tam    Usad  Disp  Uso% Montado em
/dev/sda2      2,8G   161M   2,5G    6% /
tmpfs          253M     0   253M    0% /lib/init/rw
udev           10M   136K   9,9M    2% /dev
tmpfs          253M     0   253M    0% /dev/shm
/dev/sda1      236M    21M   203M   10% /boot
/dev/sda3      2,8G    71M   2,6G    3% /home
/dev/sda7      958M    18M   892M    2% /tmp
/dev/sda5      3,3G   1,7G   1,5G   54% /usr
/dev/sda6      3,3G   393M   2,7G   13% /var
/dev/sda8      3,3G    74M   3,0G    3% /var/log
```

O comando *df* também exibe informações sobre inodes das partições através da flag *-i*. Vamos a pratica:



```
# df -ih
```

```
Sist. Arq.      Inodes   IUsados  ILivr  IUse% Montado em
/dev/sda2      179K     4,7K    175K    3% /
tmpfs          64K        3     64K    1% /lib/init/rw
udev           64K     947     63K    2% /dev
tmpfs          64K        1     64K    1% /dev/shm
/dev/sda1      61K       29     61K    1% /boot
/dev/sda7      61K       35     61K    1% /tmp
/dev/sda5      210K     67K    144K   32% /usr
/dev/sda6      210K     5,0K    205K    3% /var
/dev/sda8      210K     140    210K    1% /var/log
```



Mas o são inodes?

Informações básicas sobre seus arquivos e pastas, como permissões de acesso, identificação dos donos dos arquivos, data e hora do último acesso e alterações.

Para visualizar o espaço em disco de arquivos e diretórios, você pode usar o comando *du* (disk usage), onde exibe o espaço ocupado por arquivos e sub-diretórios.



```
# du
```

```
8      ./rc3.d
8      ./console
12     ./ld.so.conf.d
8      ./bonobo-activation
4      ./udev/.dev
76     ./udev/rules.d
96     ./udev
8      ./default/kdm.d
108    ./default
28     ./iproute2
16     ./cron.monthly
648    ./ssl/certs
8      ./ssl/private
672    ./ssl
4      ./rsyslog.d
8      ./calendar
8      ./cron.hourly
8      ./modprobe.d/arch
64     ./modprobe.d
16     ./skel
4      ./mysql/conf.d
12     ./mysql
8      ./python2.5
5276   .
```

No exemplo acima o comando foi digitado dentro do diretório */etc*. Vamos conhecer alguma das opções mais utilizadas pelo comando:

- a - Exibe o espaço ocupado por todos os arquivos ;
- b - Exibe o espaço ocupado em bytes;
- c - Faz uma totalização de todo espaço listado;
- D - Não conta links simbólicos;
- h - Exibe o espaço em formato (Kb, Mb) ao invés de usar blocos;
- k - Exibe o espaço ocupado em Kbytes;
- m - Exibe o espaço ocupado em Mbytes;

-s - Não faz calculo do espaço ocupado por sub-diretórios.

Exemplo de uso do comando `du` para calcular o espaço ocupado por um diretório:



```
# du -sh /etc
```

```
server:~# du -sh /etc
5,2M    /etc
```

Agora vamos entrar em um assunto tão importante quanto a verificação de espaço em disco, que é a checagem do sistema de arquivos. A manutenção em disco pode ser feita no Linux utilizando diversas ferramentas dependendo de qual sistema de arquivo você precisa gerenciar e/ou checar.

Os comandos que iniciam com **fsck** fazem checagem do sistema de arquivos, e assim como o **mkfs** você pode encontrar variações, isso vai depender do tipo de sistema de arquivos que seu sistema suporta. Para saber quais sistemas de arquivos você pode checar, digite `fsck` e tecle TAB 2 vezes:



```
# fsck (2x TAB)
```

```
fsck      fsck.ext3    fsck.minix   fsck.reiserfs
fsck.cramfs fsck.ext4    fsck.msdos   fsck.vfat
fsck.ext2  fsck.ext4dev fsck.nfs     fsck.xfs
```

Um detalhe na lista acima aos comandos **fsck.msdos**, **fsck.reiserfs** e **fsck.xfs** que só podem ser usados se você instalar alguns pacotes em seu sistema.



Mas como eu posso descobrir a qual pacote um comando pertence?

Vamos utilizar o comando `dpkg -S caminho_do_comando`, e para facilitar utilize um sub shell conforme o exemplo abaixo:



```
# dpkg -S $(which fsck.msdos)
dosfstools: /sbin/ fsck.msdos
```

Instale o pacote dosfstools para seu sistema ter suporte a checagem de sistema de arquivos MSDOS. Para descobrir qual pacote preciso instalar para suporte a reiserfs e xfs:



```
# dpkg -S $(which fsck.reiserfs)
reiserfsprogs: /sbin/ fsck.msdos
```



```
# dpkg -S $(which fsck.xfs)
xfsprogs: /sbin/ fsck.xfs
```

Vamos utilizar alguns comandos para fazer diagnósticos e reparos em sistema de arquivos ext2 e ext3, através dos comandos fsck.ext2, fsck.ext3 ou e2fsck.

Antes vamos criar um sistema de arquivos ext2 através do comando mke2fs:



```
# mke2fs /dev/sda10
```

A checagem segue usando o comando fsck.ext2 na partição criada, mas em partições que estão já montadas, primeiro as desmonte para não danificar os dados, através do comando umount:

Exemplo:



```
# mount | grep /home
/dev/sda3 on /home type ext3 (rw)
```

1 - Procurar qual partição que o home dos usuários está montada, e qual o sistema de arquivos usado.

2 - Demontar o ponto de montagem da partição



```
# umount /home
```

3 - Pronto! Vamos a checagem:



```
# fsck.ext3 -f /dev/sda3
ou
# e2fsck -f /dev/sda3
```

```
e2fsck 1.41.3 (12-Oct-2008)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda3: 232/183264 files (0.0% non-contiguous), 29472/732965 blocks
```

Para exibir informações de baixo nível sobre sistema de arquivos ext2 e ext3. Podemos usar os comandos `dumpe2fs` e `tune2fs` com as flags adequadas. Vamos a pratica:



```
# dumpe2fs -h /dev/sda3
```

```
dumpe2fs 1.41.3 (12-Oct-2008)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         b2c27d57-320a-43c8-81da-6a71cf4c9bd0
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype s
parse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              183264
Block count:              732965
Reserved block count:     36648
Free blocks:              703493
Free inodes:              183032
First block:              0
Block size:               4096
Fragment size:            4096
```



```
# tune2fs -l /dev/sda3
```

```
tune2fs 1.41.3 (12-Oct-2008)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         b2c27d57-320a-43c8-81da-6a71cf4c9bd0
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype s
parse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              183264
Block count:              732965
Reserved block count:     36648
Free blocks:              703493
Free inodes:              183032
First block:              0
Block size:               4096
Fragment size:            4096
```



Para converter sistema de arquivos ext2 em ext3 de uma partição, sem perda de dados use a flag -j do comando tune2fs!

Exemplo:



```
# tune2fs -j /dev/sda10
```

Para fazer alterações em baixo nível em sistema de arquivos ext2 e ext3 você pode usar o comando debugfs. Vamos a um exemplo prático onde você pode recuperar arquivos apagados através do comando debugfs:

1 - Vamos atribuir a partição /dev/sda10 o sistema de arquivos ext2:



```
# mke2fs /dev/sda10
```

2 - Crie um ponto de montagem:



```
# mkdir /media/dados
```

3 - Faça a montagem da partição:



```
# mount /dev/sda10 /media/dados
```

4 - Copie alguns dados para a partição montada:



```
# cp /etc/*.conf /media/dados
```

5 - Apague um arquivo da partição montada:



```
# rm /media/dados/adduser.conf
```

6 - Use o comando debugfs para iniciar a recuperação:



```
# debugfs -w /dev/sda10  
debugfs:
```

O debugfs abre um prompt interativo onde você pode usar comandos, que são obtidos através do help:

Available debugfs requests:

```
show_debugfs_params, params      Show debugfs parameters
open_filesys, open               Open a filesystem
close_filesys, close            Close the filesystem
feature, features                Set/print superblock features
dirty_filesys, dirty            Mark the filesystem as dirty
init_filesys                     Initialize a filesystem (DESTROYS DATA)
show_super_stats, stats         Show superblock statistics
ncheck                           Do inode->name translation
icheck                           Do block->inode translation
change_root_directory, chroot   Change root directory
change_working_directory, cd     Change working directory
list_directory, ls              List directory
show_inode_info, stat           Show inode information
link, ln                         Create directory link
unlink                           Delete a directory link
mkdir                           Create a directory
rmdir                           Remove a directory
rm                               Remove a file (unlink and kill_file, if appropriate)
kill_file                       Deallocate an inode and its blocks
```

7 - Para listar arquivos apagados use o comando **lsdel**. O detalhe do comando **debugfs** é a informação em inodes. Tecle **q** para sair.



debugfs: lsdel

```
Inode  Owner  Mode   Size   Blocks  Time deleted
  12    0 100644 2986   3/      3 Thu Aug 12 02:39:08 2010
1 deleted inodes found.
```

8 - Para recuperar o arquivo **adduser.conf**, use o comando **dump** + o número do inode apagado e associe ao nome do arquivo:



debugfs: dump <12> /media/dados/adduser.conf

Use o comando **lsdel** para confirmar se o arquivo foi recuperado, e o comando **quit** para sair do prompt interativo.

Capítulo 2

Gerenciando

2.1. Objetivos

- Troubleshooting: Funcionamento dos comandos e2fsck e resize2fs.

2.1. Troubleshooting



Como posso verificar a integridade do sistema de arquivos em meu HD?

Quando o sistema de arquivos por algum motivo apresenta falhas ao iniciar a máquina, devido ao desligamento incorreto do sistema, falha de energia ou a presença de badblocks, é possível fazer uma verificação através do comando e2fsck.

Como exemplo vamos verificar de no modo verbose e badbloks a integridade do sistema de arquivos. Não esqueça de antes desmontar a partição:



```
# e2fsck -fvc /dev/sdb1
```



Como posso diminuir o tamanho de uma partição sem perder os dados?

Isso é possível através do comando `resize2fs`, onde você pode redimensionar sistema de arquivos `ext2`, `ext3` ou `ext4`. A operação de redimensionamento deve ser feita com a partição desmontada. Se você precisa redimensionar uma partição que o sistema esta utilizando, reinicie a maquina com um Live CD.

O programa `resize2fs` não manipula o tamanho das partições, e sim do sistema de arquivos. Então você deve primeiro usar algum gerenciador de partição, como `fdisk` ou `cfdisk` se precisar aumentar uma partição. Com isso seus dados serão apagados, ao menos que você utilize LVM

Em nosso exemplo a partição terá seu tamanho reduzido, assim criando uma nova partição menor. Quando você reduz o tamanho de uma partição, seus dados não são apagados. Vamos a prática:

Em nosso cenário uma partição de 10GB que tem 808MB ocupado e 8.6GB livres. A nossa missão é reduzir a partição para 6GB, assim ganhando uma nova partição de 4GB.

Esses valores são apenas para mostrar os comandos, apenas substitua por valores de seu HD.

Vamos usar o comando `mount` e `df -h` para mostrar o estado atual da partição.



```
# mount | grep /dev/sdb1 ; df -h | grep /dev/sdb1
```

```
/dev/sdb1 on /media/dados type ext3 (rw)
/dev/sdb1          9,9G  808M  8,6G   9% /media/dados
```

Como podemos ver em nosso exemplo a partição `/dev/sdb1` está montada e usando o sistema de arquivos `ext3`. Vamos ao passo a passo:

1 - Desmonte a partição que você ira redimensionar:



```
# umount /dev/sdb1
```

2 - Faça a verificação do sistema de arquivos:



```
# fsck -n /dev/sdb1
```

3 - Converta a partição para ext2:



```
# tune2fs -O ^has_journal /dev/sdb1
```

4 - Novamente faça a verificação do sistema de arquivos:



```
# e2fsck -f /dev/sdb1
```

```
e2fsck 1.41.3 (12-Oct-2008)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb1: 10304/655360 files (0.1% non-contiguous), 215479/2620595 blocks
```

5 - Redimensione a partição com o comando `resize2fs` para 6000MB, lembrando que a partição atualmente tem 808MB usados:



```
# resize2fs /dev/sdb1 6000M
```

```
resize2fs 1.41.3 (12-Oct-2008)
Resizing the filesystem on /dev/sdb1 to 1536000 (4k) blocks.
The filesystem on /dev/sdb1 is now 1536000 blocks long.
```

6 - A próxima etapa é diminuir o tamanho da partição, anote o valor apresentado no comando acima (1536000). Use o comando `fdisk` e tecle `p` para listar tabela de partições:



```
# fdisk /dev/sdb1
```

```
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000
```

Dispositivo	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	1305	10482381	83	Linux

7 - A próxima etapa é excluir a partição de 10GB e criar 2 partições. A primeira com 6Gb e a segunda 4GB, tecle `d` para excluir a partição:

```
Comando (m para ajuda): d
Selected partition 1
```

8 - Para criar a primeira partição, tecle `n` (new), tecle `p` para indicar partição primaria, `1` para indicar que é a primeira partição, e `1` primeiro cilindro:

```
Comando (m para ajuda): n
Comando - ação
    e   estendida
    p   partição primária (1-4)
p
Número da partição (1-4): 1
Primeiro cilindro (1-1305, default 1): 1
```

9 - Nesta etapa vamos digitar o valor do ultimo cilindro usado uma formula:

$$1536000 * 4 + 307200 = 6465200$$

Onde:

1536000 - Quantidade de blocos da saída do comando `resize2fs` (1536000);

4 - Tamanho de um bloco (4k);

307260 - 5% do resultado da conta acima $1536000 * 4$

Digite o resultado da formula com o sinal de + e a letra K ao final:

```
Último cilindro ou +tamanho ou +tamanho M ou +tamanho K (1-1305, default 1305):
+6451200K
```

10 - Tecle w para gravar as alterações na tabela de particionamento:

```
Comando (m para ajuda): w
A tabela de partições foi alterada!

Chamando ioctl() para reler tabela de partições.
Sincronizando discos.
```

11 - Reinicie a maquina e ao retornar, faça a checagem na partição redimensionada:



```
# fsck -n /dev/sdb1
```

12 - Use o comando `tune2fs` com a flag -j para converter a partição ext2 em ext3 sem perder os dados:



```
# tune2fs -j /dev/sdb1
```

```
tune2fs 1.41.3 (12-Oct-2008)
Creating journal inode: done
This filesystem will be automatically checked every 28 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

13 - Faça a montagem e verifique com o comando `df -h` o novo tamanho da partição:



```
# mount -t ext3 /dev/sdb1 /media/dados
```



```
# df -h | grep /dev/sdb1
```

Agora você pode acessar seus arquivos da partição redimensionada, sem ter nenhuma perda de dados, e o novo espaço de 4GB pode ser utilizado.



```
# cfdisk /dev/sdb
```

cfdisk (util-linux-ng 2.13.1.1)					
Disco: /dev/sdb					
Size: 10737418240 bytes, 10.7 GB					
Heads: 255 Sectors per Track: 63 Cylinders: 1305					
Nome	Opções	Tipo Part.	Tipo S/A	[Rótulo]	Size (MB)
sdb1		Primária	Linux	ext3	5613,13
		Pri/lóg	Espaço livre		4120,87