



Restaurando partições danificadas com o *dd_rescue*

Operação resgate!

Se você faz backup regular de seus dados, não precisa esquentar a cabeça caso seu disco rígido vá pro bebelê. Entretanto, o que aconteceria se você esquecesse, um único dia, o importantíssimo backup? A lei de Murphy afirma que é exatamente nesse dia que o disco apresentará erros. Sorte sua se o “*dd_rescue*” estiver em sua maleta de primeiros socorros.

POR HANS-GEORG ESSER

Há um sem-número de causas potenciais para uma falha no HD. Você irá precisar dos serviços de um bureau de recuperação de discos rígidos se uma pane qualquer destruir as cabeças de gravação e leitura.

Às vezes o dano é menor: apenas alguns setores podem estar ilegíveis e o sistema interno de recuperação do disco, que deveria ter setores de reserva para esses casos, pode estar lotado.

Com um pouco de sorte, você pode ainda ser capaz de montar as partições em um HD danificado e as mensagens de erro ficarem restritas a alguns arquivos (ou apenas um, com uma pitada a mais de sorte). Nesse caso você pode só copiar os arquivos da partição danificada em um novo disco rígido. Quando terminar, pode simplesmente descartar o disco ou substituí-lo, se ainda estiver na garantia. O que quer que faça, evite usar o disco danificado. Embora seja possível usar marcas em setores defeituosos com a ferramenta *badblocks* (veja o quadro

1) para, explicitamente, escondê-los, o risco do disco literalmente “morrer nos seus braços” é enorme.

Falha de montagem

Os desesperadores momentos em que você não consegue montar uma partição são, de longe, muito mais sérios. O comando *mount* falhará se os dados de gerenciamento de disco estiverem danificados e o *fsck* não for capaz de reverter a situação – afinal, é impossível escrever em setores ruins (veja quadro 2).

Só há uma opção: criar uma cópia do sistema de arquivos e armazená-la em

Quadro 1: Procurando setores defeituosos

Como o nome sugere, a ferramenta *badblocks* procura por setores defeituosos em mídias magnéticas – notadamente disquetes e discos rígidos. Você pode usar o *badblocks* em conjunto com o *fsck* para marcar setores danificados, evitando seu uso pelo sistema de arquivos. Para isso, passe a saída do comando *badblocks* à entrada do *fsck*:

```
badblocks /dev/hdb5 > /tmp/bad-z
blocks
fsck -t ext2 -l /tmp/bad-blocks z
/dev/hdb5
```

Ao reformatar um disco danificado, não é preciso chamar o *badblocks* à luta. O comando *mkfs* possui a opção *-c* (check), que automaticamente chama o *badblocks* antes da formatação:

```
mke3fs -c /dev/hdb5
```

Quadro 2: Erros de disco

Os erros de leitura que impedem a montagem de uma partição são indicados como tal pelo kernel em seu registro de eventos (log). As advertências são expressas da seguinte maneira:

```
Feb 6 05:28:42 server kernel: reiserfs: found format “3.6” with standard journal
Feb 6 05:28:47 server kernel: reiserfs: enabling write barrier flush mode
Feb 6 05:28:47 server kernel: reiserfs: using ordered data mode
Feb 6 05:28:50 server kernel: hdb: dma_intr: status=0x51 z
{ DriveReady SeekComplete Error }
Feb 6 05:28:50 server kernel: hdb: dma_intr: error=0x40 z
{ UncorrectableError }, LBAsect=19616787, sector=65680
Feb 6 05:28:50 server kernel: end_request: I/O error, dev 03:42 (hdb), sector 65680
Feb 6 05:28:50 server kernel: journal-459: unable to read journal header
Feb 6 05:28:50 server kernel: sh-2022: reiserfs_read_super: unable to initialize z
journal space
```

um segundo disco. Tente, então, modificar a imagem gerada usando o `fsck`. Como os sistemas de arquivos possuem cópias dos dados de gerenciamento de disco em vários lugares, é possível recuperar seus dados.

Knoppix: `dd_rescue` a bordo

O `dd_rescue`, ferramenta desenvolvida por Kurt Garloff [1], é uma variação de uma velha ferramenta do Unix, o `dd`. O `dd_rescue` foi idealizado com a recuperação de dados em mente. O Knoppix inclui o `dd_rescue`, portanto seus usuários não precisam compilar o programa a partir do código fonte. Usuários de outros sistemas operacionais

devem consultar o site do `dd_rescue` para mais informações. Neste artigo, tomaremos como base o Knoppix.

Além da facilidade de uso, o `dd_rescue` possui duas grandes vantagens em relação ao `dd`:

- Se um erro ocorrer durante a cópia, o `dd_rescue` não interromperá a cópia, simplesmente. Em vez disso, copia para o destino uma cadeia de zeros com o mesmo tamanho do setor defeituoso. Isso resulta numa imagem completa da partição, apesar dos bytes nulos nos setores defeituosos.
- O `dd_rescue` pode usar dois tamanhos de bloco para acesso de leitura. Nas áreas sem defeitos, lê blocos cujo tamanho padrão é de 16384 bytes; em caso de erro, a ferramenta reduz o tamanho do bloco para 512 bytes. Ambos os valores são configuráveis.

Preparação

Em primeiro lugar, é preciso instalar o disco rígido em uma máquina. Também é preciso outro disco com uma partição livre e grande o bastante para comportar os próximos passos. A nova partição

deve possuir o dobro da capacidade da partição do disco danificado que queremos recuperar.

Depois de iniciar o computador com o Knoppix, abra um console e alterne para usuário `root` com o comando `sudo su` (não é necessário senha). Então rode `fdisk -l` para obter uma lista de partições.

Depois de identificar a partição danificada, a operação de recuperação pode ter início. Primeiro, monte uma partição no segundo disco rígido (o que não contém erros) em `/mnt`, por exemplo:

```
mount /dev/hdb1 /mnt
```

Assumindo que `/dev/hda5` é a partição danificada, o comando apropriado é:

```
dd_rescue /dev/hda5 /mnt/👉  
imagem.dat
```

Com o `dd_rescue`, não há necessidade dos parâmetros típicos do `dd`, `if=` e `of=`. O primeiro parâmetro sempre especifica a origem e o segundo o destino. O programa mostra uma barra de

GLOSSÁRIO

dd: A velha e boa ferramenta “dd”, herdada dos Unix tradicionais, copia arquivos entre dispositivos de bloco e arquivos comuns.

Pode ser usada, por exemplo, para copiar uma imagem em um disquete (os desenvolvedores provavelmente teriam preferido chamar a ferramenta de “cc” – copy and convert – mas esse mnemônico já estava em uso pelo compilador C).

```

Session Edit View Bookmarks Settings Help
linux:~# dd_rescue -h
dd_rescue Version 1.02, garloff@use.de, GNU GPL
($Id: dd_rescue.c,v 1.39 2001/07/24 12:08:10 garloff Exp $)
dd_rescue copies data from one file (or block device) to another
USAGE: dd_rescue [options] infile outfile
Options: -s ipos    start position in input file (default=0),
-S opos    start position in output file (def=ipos);
-b softbs  block size for copy operation (def=16384),
-B hardbs  fallback block size in case of error (def=512);
-e maxerr  exit after maxerr errors (def=0=infinite);
-m maxxfer maximum amount of data to be transferred (def=0=inf);
-l logfile name of a file to log errors and summary to (def="");
-r        reverse direction copy (def=forward);
-t        truncate output file (def=no);
-w        abort on write errors (def=no);
-a        sparse file writing (def=no),
-A        always write blocks, zeroed if err (def=no);
-i        interactive: ask before overwriting data (def=no);
-f        force: skip some sanity checks (def=no);
-q        quiet operation;
-v        verbose operation;
-V        display version and exit;
-h        display this help and exit.
Note: Sizes may be given in units b(=512), k(=1024), M(=1024^2) or G(=1024^3) bytes
This program is useful to rescue data in case of I/O errors, because
it does not necessarily abort or truncate the output.
linux:~#

```

Figura 1: A opção “-h” do “dd_rescue” mostra sua lista de opções.

progresso enquanto está em execução e emite advertências sempre que um erro é descoberto. O quadro 3 mostra uma sessão típica do `dd_rescue`, incluindo a sintaxe do comando e os resultados.

Nos casos em que o dano no disco é mínimo, em poucos minutos uma imagem quase perfeita é criada. Essa imagem terá todos os dados legíveis nas mesmas posições que a imagem original.

Juntando os cacos

Se uma área maior do disco foi pra cucuia, o `dd_rescue` pode não conseguir passar de um certo ponto com um grande número de setores defeituosos. Nesse caso, a única opção é interromper o processo e executar o `dd_rescue` novamente, usando um arquivo de destino diferente e estipulando a opção `-r`:

```
dd_rescue -r /dev/hda5
/mnt/imagem-fim.dat
```

Quadro 3: uma sessão com o dd_rescue

```

root@tty0[knoppix]# dd_rescue /dev/hda5 /mnt/imagem.dat
dd_rescue: (info): ipos:      859444.5k, opos:      859444.5k, xferd:      859444.5k
                errs:       0, errxfer:      0.0k, succxfer: 859444.5k
                +curr.rate: 30166kB/s, avg.rate: 41084kB/s, avg.load: -0.6%
dd_rescue: (warning): /dev/hda5 (859444.5k): Input/output error!
dd_rescue: (info): ipos:      80043264.0k, opos:     80043264.0k, xferd:     80043264.0k
                errs:       1, errxfer:      0.5k, succxfer: 80043263.5k
                +curr.rate: 30166kB/s, avg.rate: 41084kB/s, avg.load: -0.6%
dd_rescue: (info): /dev/hda5 (80043264.0k): EOF
Summary for /dev/hda5 -> /mnt/imagem.dat:
dd_rescue: (info): ipos:      80043264.0k, opos:     80043264.0k, xferd:     80043264.0k
                errs:       0, errxfer:      0.5k, succxfer: 80043263.5k
                +curr.rate:      0kB/s, avg.rate: 41084kB/s, avg.load: -0.6%
root@tty0[knoppix]# _

```

Com isso, o `dd_rescue` inicia a cópia pelo fim da partição e trabalha de trás para a frente até chegar nos setores ruins. É claro que, tendo chegado aos mesmos setores defeituosos de antes, provavelmente o programa ficará “patinando” e será preciso interrompê-lo novamente. Com-

parando os tamanhos dos dois arquivos, `imagem.dat` e `imagem-fim.dat`, com o tamanho do arquivo original, é possível descobrir o tamanho da seção faltante. Para criar um arquivo preenchido com zeros do tamanho apropriado, digite:

```
dd if=/dev/zero of=/mnt/
imagem-zero.dat bs=1024 count=N
```

e cole cada um dos pedaços para formar uma única imagem:

```
cd /mnt; cat imagem.dat imagem-
zero.dat imagem-fim.dat >
imagem-total.dat
```

Reparação

Antes de começar a trabalhar na imagem que o `dd_rescue` criou, faça uma cópia de segurança usando o comando `cp`. É por isso que recomendo um disco com o dobro do espaço da partição danificada:

```
cp /mnt/imagem.dat
/mnt/imagem.dat.copy
```

Você pode usar o `fsck` com as opções apropriadas para reparar os arquivos de imagem. Para uma partição ReiserFS, o comando seria:

```
reiserfsck -fix-fixable
/mnt/imagem.dat
```

ou

```
fsck.ext3 -p /mnt/imagem.dat
```

para um sistema de arquivos Ext3.

Depois, monte o sistema reparado para certificar-se de que tudo está bem (monte como “read only”):

```
mkdir /temp
mount -o loop,ro /mnt/
imagem.dat /temp
```

Agora deve ser possível acessar os dados na partição danificada, montada no diretório `/temp`, e copiar tudo o que for necessário para outros diretórios:

```
cp -a /temp diretorio_destino
```

Como alternativa, é possível usar:

```
rsync -av -rsh="ssh"
/temp root@computer:diretorio_
destino/
```

para enviar os arquivos para outro computador na rede. ■

SOBRE O AUTOR

Hans-Georg Esser é autor dos livros SuSE Linux zum Nachschlagen, Das große Buch - Linux e Linux Intern, publicados na Alemanha, além de vários artigos publicados na revista Linux User. Atualmente, é editor da revista Easy Linux.



INFORMAÇÕES

- [1] Página oficial do `dd_rescue`: <http://www.garloff.de/kurt/linux/ddrescue/>
- [2] Knoppix: <http://knopper.net/knoppix>
- [3] Artigo sobre o DD na Wikipedia: [http://en.wikipedia.org/wiki/Dd_\(Unix\)](http://en.wikipedia.org/wiki/Dd_(Unix))