

Capítulo 31

Compactação de arquivos

A compactação de arquivos é uma tarefa essencial que deve ser bem conhecida pelo administrador do sistema. Este capítulo irá abordar vários padrões de compactação utilizados no GNU/Linux.

31.1 Padrões de compactação

Como será que o PKZIP e o WinZIP trabalham no MS-DOS e no MS Windows? Como eles reúnem vários arquivos e geram um novo arquivo de tamanho menor? Na verdade, esses aplicativos executam duas tarefas básicas: o empacotamento e a compactação.

O empacotamento é o ato de reunir vários arquivos e colocá-los dentro de um outro, denominado arquivo pacote. Esse pacote possui um tamanho final próximo à soma dos arquivos que o compõe.

Na verdade, o tamanho ficará ligeiramente menor em virtude de ser um único arquivo e, em consequência, ocupar menos inodes etc. Isso gera um arquivamento menor de informações referentes aos metadados. Mas não houve ainda uma operação de compactação.

A compactação é o ato de realizar uma compressão de dados, atuando sobre seqüências de bits repetidos existentes em um arquivo ou pacote.

A compactação poderá ser realizada em um pacote (conjunto de arquivos) ou em um arquivo comum.

No GNU/Linux existem vários empacotadores e compactadores. O utilitário empacotador mais utilizado é o `tar`. Os compactadores mais comuns são o `compress`, o `gzip` e o `bzip2`. Também há o `zip`, que é totalmente compatível com o WINZIP.

O `zip` realiza o empacotamento e a compactação ao mesmo tempo.

As extensões resultantes são as seguintes:

Extensão	Agente empacotador e/ou compactador
.tar	Empacotador tar.
.Z	Compactador compress.
.gz	Compactador gzip.
.bz2	Compactador bzip2.
.zip	Empacotador e compactador zip.
.tar.Z	Empacotador tar + compactador compress.
.tar.gz	Empacotador tar + compactador gzip.
.tar.bz2	Empacotador tar + compactador bz2.

31.2 Empacotamento com o tar

A sintaxe para realizar o empacotamento com o tar é a seguinte:

```
# tar -cvf <arquivo_final> <origem>
```

Os arquivos gerados com o comando tar são conhecidos como tarball. A chave f deverá ser a última, uma vez que define o nome do arquivo a ser gerado.

As chaves utilizadas foram as seguintes:

Chave	Função
-c	Create. Cria um novo arquivo.
-v	Verbose. Exibe mensagens relativas ao andamento da operação.
-f	File. Define que será gerado um arquivo com nome <arquivo_final>.
<origem>	Representa o(s) arquivo(s) e/ou diretório(s) a ser(em) empacotado(s). Os arquivos a serem empacotados podem ser definidos por curingas, como o caractere asterisco (vide Capítulo 16).

O comando tar é, por padrão, recursivo. Por exemplo: se estivermos dentro de um diretório e desejarmos empacotar todos os seus arquivos e subdiretórios, gerando um arquivo chamado teste.tar, utilizaremos o comando:

```
# tar -cvf teste.tar *
```

Para empacotarmos todos os arquivos .gif existentes no diretório /var/www/html e colocarmos o resultado dentro do arquivo gifs.tar, no diretório /tmp, utilizaremos o comando:

```
# tar -cvf /tmp/gifs.tar /var/www/html/*.gif
```

Lembre-se que, em muitas distribuições, o conteúdo do diretório /tmp é totalmente apagado a cada reinicialização.

31.3 Desempacotamento com o tar

O desempacotamento é muito semelhante ao empacotamento. Veja a sintaxe:

```
# tar -xvf <arquivo a ser desempacotado>
```

A única novidade é a opção `x` no lugar da opção `c`. A opção `x` desempacota o arquivo declarado com a opção `f`.

31.4 Compactação com compress

```
Instale com: # apt-get install ncompress
```

Qualquer arquivo poderá ser compactado com o comando `compress`, seja ele um arquivo comum ou um pacote (gerado pelo `tar`). A sintaxe do `compress` é a seguinte:

```
# compress <arquivo>
```

Exemplo:

```
# compress teste.tar
```

Será gerado o arquivo `teste.tar.Z`.

31.5 Descompactação com uncompress

O comando `uncompress` irá descompactar os arquivos compactados com comando `compress`. Exemplo:

```
# uncompress teste.tar.Z
```

O comando anterior resultará no arquivo `teste.tar`.

31.6 Compactação com gzip

Para compactar um arquivo com o `gzip`, utilizaremos a seguinte sintaxe:

```
# gzip <arquivo>
```

Exemplo:

```
# gzip teste.tar
```

Será gerado o arquivo `teste.tar.gz`.

O `gzip` é o compactador GNU/Linux mais utilizado para a disponibilização de arquivos na Internet.

31.7 Descompactação com gunzip

O `gunzip` irá descompactar tudo o que foi compactado com `gzip`. Exemplo:

```
# gunzip teste.tar.gz
```

O resultado será o arquivo `teste.tar`.

31.8 Mesclando o tar e gzip

O GNU/Linux oferece a possibilidade de mesclarmos os comandos `tar` e `gzip` pelo simples acréscimo da chave `z` no comando `tar`. Assim, para empacotar com `tar` e comprimir com `gzip`, numa só operação, utilize:

```
# tar -cvzf <arquivo_final> <origem>
```

A descompressão e o desempacotamento poderão ser feitos com o seguinte comando:

```
# tar -xvzf <arquivo>
```

Se utilizar "Z", em vez de "z", haverá uma mesclagem com compress.

31.9 bzip2 e bunzip2

Instale com: `# apt-get install bzip2`

A utilização do `bzip2` e do `bunzip2` é similar à do `gzip/gunzip`. O `bzip2` alcança altas taxas de compressão. Exemplos:

```
# bzip2 teste.tar
```

```
# bunzip2 teste.tar.bz2
```

31.10 Mesclando o tar e bzip

Similar ao que ocorre com o `gzip`, o `bzip2` pode ser executado juntamente com o comando `tar`, bastando acrescentar a opção `j`. Exemplos:

```
# tar -cvjf <arquivo_final> <origem>
```

```
# tar -xvjf <arquivo>
```

31.11 zip e unzip

Instale com: `# apt-get install zip unzip`

Para utilizar o `zip`, adote a seguinte sintaxe:

```
# zip <arquivo a ser gerado> <arquivos origem>
```

Exemplo:

```
# zip /tmp/etc.zip /etc/*
```

Diferentemente do `tar`, o `zip` não é recursivo por padrão. Se desejar compactação recursiva, utilize as chaves `-r` e `-p`. Exemplo:

```
# zip /tmp/etc.zip -r -p /etc/*
```

As chaves citadas só funcionam separadamente.

Para descompactar um arquivo `.zip`, utilize o comando `unzip`. Exemplo:

```
# unzip etc.zip
```

31.12 Comparação entre os compactadores

A partir de agora, faremos três comparações simples entre os compactadores. A primeira será relativa ao poder de compressão. A segunda versará sobre o tempo gasto para comprimir. A terceira terá como objeto o tempo gasto para descomprimir.

Os testes que serão mostrados a seguir ocorreram em uma máquina Celeron 2.2, com 384 MB de memória RAM e HD ATA 100. O arquivo utilizado para testes era único, continha caracteres aleatórios e o seu tamanho era de 50 MB. O filesystem utilizado foi o ReiserFS.

31.12.1 Poder de compressão

O teste de compressão produziu os resultados mostrados na Figura 31.1. Quanto maior a barra, maior foi o percentual de compressão.

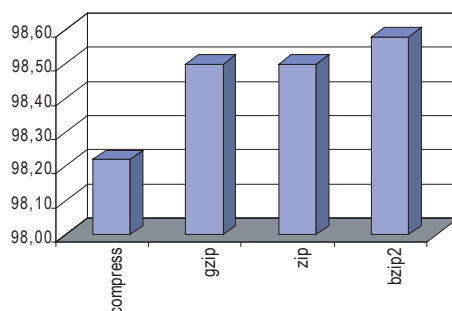


Figura 31.1 – Percentual de compressão obtido.

É possível notar que o `bzip2` tem um bom poder de compressão. No entanto, todos os compressores conseguiram compactar o arquivo, que era somente texto, em mais de 95%.

31.12.2 Tempo para comprimir

O teste relativo ao tempo de compressão produziu os resultados mostrados na Figura 31.2.

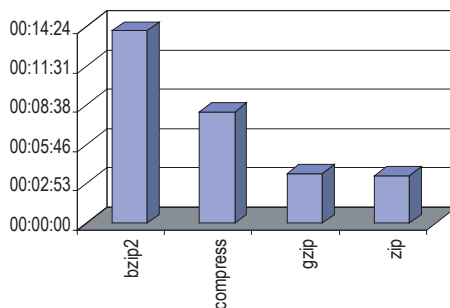


Figura 31.2 – Tempo gasto para comprimir.

Nesse teste, o `bzip2` perdeu disparado em relação aos outros. Com isso, pode-se afirmar que o `gzip` é o compactador que apresenta o melhor custo/benefício.

31.12.3 Tempo para descomprimir

O teste relativo ao tempo de descompressão produziu os resultados mostrados na Figura 31.3.

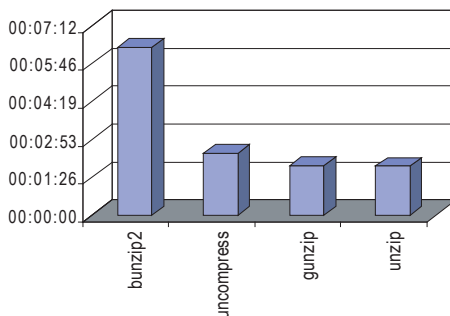


Figura 31.3 – Tempo gasto para descomprimir.

O `bunzip2`, parceiro do `bzip2`, teve o pior desempenho.

31.13 Outros compactadores

Existem outros compactadores. Podemos citar o `lha`, o `arj` (velho conhecido de quem é oriundo do MS-DOS), e o `ar`. O `ar` é o compactador utilizado para criar os pacotes `.deb`. Ou seja: os arquivos `.deb` não passam de pacotes compactados com `ar`. Para descompactar um arquivo `.deb`, utilize:

```
# ar -x arquivo.deb
```

31.14 Conclusão

Um arquivo “compactado” geralmente é resultante de uma operação de empacotamento seguida de uma compressão. Geralmente, o empacotamento é feito com o utilitário `tar`. O `tar` (tape archive) foi desenvolvido, originalmente, para fazer backups seqüenciais em fitas.

Quanto aos compactadores, existem vários deles. Cada um tem vantagens e desvantagens. O compactador mais utilizado para disponibilizar códigos-fonte na Internet é o `gzip`, seguido do `bzip2`.

Apenas como curiosidade, no MS Windows, os Winzips atuais manipularam arquivos `tar`, `gzip` e `compress`. O Filzip, que é freeware, também suporta esses formatos. O Filzip está disponível em <http://www.filzip.com/en/index.html>.