



Compactação e Descompactação de Arquivos

Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro

Prof. Edwar Saliba Júnior

Abril de 2025



Compactação de Arquivos

- O que significa compactar um arquivo?
- E como podemos fazer a compactação de um arquivo?



Exemplo de Compactação de Arquivos por Repetição de Palavras

- Dado o texto a seguir, vamos identificar as palavras com mais de uma letra e que possuem repetições no texto:
 - A casa rosa em cima da montanha estava cheia de flores. Havia flores nas cores: vermelha, rosa, branca e amarela. O tom de rosa das flores era vibrante e realçava a cor da casa da montanha.



Exemplo de Compactação de Arquivos por Repetição de Palavras

- Agora vamos criar uma tabela com as palavras que se repetem:

Símbolo	Palavra	Repetições	Qtd. Letras	Economia
1	casa	2	4	8
2	rosa	3	4	12
3	da	4	2	8
4	montanha	2	8	16
5	flores	3	6	18
6	cor	2	3	6



Exemplo de Compactação de Arquivos por Repetição de Palavras

- A frase completa possui 191 caracteres.
- Vamos assumir que:
 - cada caractere possua o tamanho de 1 byte,
 - vamos desconsiderar os dados de controle (cabeçalho) do arquivo,
 - então nosso arquivo terá o tamanho de 191 bytes.



Exemplo de Compactação de Arquivos por Repetição de Palavras

- Agora vamos simular uma compactação de arquivo, pegando as palavras que se repetem, descritas na tabela anterior e
- substituindo-as no texto pelo “símbolo” correspondente à palavra na tabela.
- E assim temos:



Exemplo de Compactação de Arquivos por Repetição de Palavras

- Tabela:
- Texto modificado:

Símbolo	Palavra
1	casa
2	rosa
3	da
4	montanha
5	flores
6	cor

– A 1 2 em cima 3 4 estava cheia de 5. Haviam 5 nas 6es: vermelha, 2, branca e amarela. O tom de 2 3s 5 era vibrante e realçava a 6 3 1 3 4.

Se substituirmos os números (símbolos) pela sua palavra correspondente na tabela, teremos o texto original novamente.

Texto compactado. Ele tem 139 caracteres.



Exemplo de Compactação de Arquivos por Repetição de Palavras

- Conta rápida:
 - tam. do texto original: **191**
 - tam. do texto compactado: **139**
$$191 - 139 = \underline{52}$$
- Então economizamos 52 bytes?
 - Não exatamente!
- Veja! Para que o *software* que descompacta o arquivo consiga retorná-lo ao que ele era antes da compactação, é necessário que a tabela criada seja anexada ao arquivo compactado.



Exemplo de Compactação de Arquivos por Repetição de Palavras

Símbolo	Palavra
1	casa
2	rosa
3	da
4	montanha
5	flores
6	cor

- Então temos que descontar o tamanho da tabela (que será anexada ao arquivo).
- Se anexarmos algo do tipo: "1 casa 2 rosa 3 da..." esta tabela terá o tamanho de 45 caracteres.
- Então, refazendo a conta temos:

$$191-139+45 = \underline{7}$$



Exemplo de Compactação de Arquivos por Repetição de Palavras

- Conseguimos então, com a compactação, economizar apenas **7 bytes**.
- Não se assuste! Isto costuma acontecer quando o arquivo é muito pequeno e usando-se este tipo de compactação.
- Existem casos também onde a compactação pode exercer um efeito inverso. Ou seja, ao invés de diminuir o tamanho do arquivo ao se compactar, aumenta-se! Mas, esta situação costuma ocorrer somente com arquivos pequenos.



Compactação e Descompactação

- Agora que você já tem uma ideia de como funciona a compactação de arquivos, vamos conhecer alguns compactadores.
- O Linux possui diversos utilitários para compactação de arquivos. Dentre eles podemos citar:
 - tar,
 - tar.gz
 - tar.bz2
 - gzip/gunzip
 - zip/unzip



Comando tar

- O comando `tar` é muito utilizado para agrupar vários arquivos em um só (sem compressão de dados). Isto é usado em situações distintas, por exemplo, quando se tem que transferir de vários arquivos via rede. Ao invés de executar vários comandos ou um comando utilizando curingas, então, cria-se um único arquivo com o comando `tar` e faz-se a transferência.
- O comando `tar` possui diversos parâmetros, dentre eles vejamos 4 que nos serão bem úteis:
 - c cria um novo arquivo ".tar" e adiciona os arquivos a serem agrupados;
 - f indica que o destino é um arquivo em disco;
 - v exibe o nome de cada arquivo que comporá o novo na junção e
 - x extrai os arquivos agrupados no arquivo ".tar".



Comando tar

- Como explicado no *slide* anterior, o comando `tar` é muito utilizado para agrupar arquivos em um só, sem compressão de dados, Vamos tatar:

```
$ tar cf backup.tar agenda titulo.txt
```

- o comando acima cria o arquivo `backup.tar` sem compressão de dados.
- Para visualizarmos o conteúdo do arquivo `backup.tar` podemos utilizar o comando:

```
$ tar tvf backup.tar
```

- o comando listará os arquivos contidos no arquivo `backup.tar`.
- E para descompactar o arquivo `backup.tar` nós utilizamos o comando:

```
$ tar xvf backup.tar
```



Comando `tar.gz`

- O comando `tar.gz` consiste em 2 processos: o `tar` para agrupar os arquivos e o `gz` para fazer a compactação dos dados. Por exemplo, vamos compactar os arquivos que criamos:

```
$ tar zcvf backup.tar.gz agenda titulo.txt
```

- o comando acima cria o arquivo `backup.tar.gz` com compressão de dados.
- Para visualizarmos o conteúdo do arquivo `backup.tar.gz` podemos utilizar o comando:

```
$ tar tvf backup.tar.gz
```

- e para descompactar o arquivo `backup.tar.gz` nós utilizamos o comando:

```
$ tar zxvf backup.tar.gz
```



Comando `tar.bz2`

- O comando `tar.bz2` também consiste em 2 processos: o `tar` para agrupar os arquivos e o `bz2` para fazer a compactação dos dados. E aqui destaca-se que a compactação do `bz2` é mais eficiente do que o `gz`, no entanto ela é mais lenta. Por exemplo, vamos compactar os arquivos que criamos:

```
$ tar jcvf backup.tar.bz2 agenda titulo.txt
```

- o comando acima cria o arquivo `backup.tar.bz2` com compressão de dados.
- Para visualizarmos o conteúdo do arquivo `backup.tar.bz2` podemos utilizar o comando:

```
$ tar tvf backup.tar.bz2
```

- E para descompactar o arquivo `backup.tar.bz2` nós utilizamos o comando:

```
$ tar jxvf backup.tar.bz2
```

- **observação:** a compactação `bz2` só mostrará sua eficiência quando aplicada a arquivos maiores. Em arquivos pequenos a compactação utilizando-se o `bz2` pode ser igual ou, até mesmo, menos eficiente que a compactação utilizando o `gz`.



Comando `gzip` e `gunzip`

- O comando `gzip` faz a compactação de um único arquivo. Por exemplo, vamos compactar os arquivos que criamos:

```
$ gzip -c -r agenda titulo.txt > backup.gz
```

- o comando acima cria o arquivo `backup.gz` com compressão de dados.
- E para descompactar o arquivo `backup.gz` nós utilizamos o comando:

```
$ gunzip -N backup.gz
```

- **observação 1:** se você tentar compactar mais de um arquivo utilizando o comando `gzip`, estes arquivos serão concatenados.
- **observação 2:** ao executar o comando `gunzip`, o parâmetro `-N` força o descompactador a restaurar o arquivo com o seu nome original. E, por fim, ao finalizar a descompactação, o comando `gunzip` apaga o arquivo que estava compactado.



Comando zip e unzip

- O comando `zip` faz a compactação de vários arquivos. Por exemplo, para compactar os arquivos que criamos:

```
$ zip backup.zip agenda titulo.txt
```

- o comando acima cria o arquivo `backup.gz` com compressão de dados.
- E para descompactar o arquivo `backup.gz` nós utilizamos o comando:

```
$ unzip backup.zip
```

- **observação:** o comando `zip/unzip` não vem instalado por padrão em todas as distribuições Linux.



Referências

- GUIA FOCA GNU/Linux. **Iniciante**. Disponível em: <<http://www.guiafoca.org/cgs/guia/iniciante/ch-intro.html>>. Acesso em: 27 jul. 2017.
- MOTA FILHO, J. E. **Descobrimo o Linux**. 3. ed. São Paulo: Novatec, 2012.
- ROMERO, D. **Começando com Linux**: Comandos, serviços e administração. São Paulo: Casa do Código, 2013.
- TRYBE. **Como localizar arquivos no Linux**. Disponível em: <https://blog.betrybe.com/localizar-arquivos-no-linux/>. Acesso em: 24 mar. 2025.