

Compresión de Archivos

gzip

Este comando:

- Recibe como parámetro un archivo
- O comprime
- Guarda o archivo comprimido coa extensión .gz
- Borra o archivo orixinal non comprimido.

Exemplo:

```
ls -l proba
-rw-r--r-- 1 pepe curso 312996 Jan 30 21:44 proba

gzip proba      → Comprime arquivo
ls -l
-rw-r--r-- 1 pepe curso 103441 Jan 30 21:45 proba.gz

gzip -l proba.gz → Amosa ratio de compresión acadado
      compressed      uncompr.      ratio uncompressed_name
      103441      312996      67%      proba

gunzip fichero.gz o gzip -d fichero.gz → Descomprime o arquivo
```

gzip -l proba → Máis rápido, pero menor tasa de compresión
gzip -9 proba → Menos rápido, pero maior tasa de compresión

tar (tape archive)

- **Problema:** gzip só pode comprimir un arquivo de cada vez.
- **Solución:** tar.

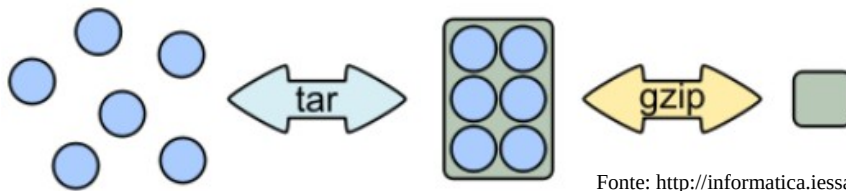
Este comando **empaqueta** un conxunto de arquivos nun só.

Tar non comprime, pero pode empregarse conxuntamente con outros comandos (bzip2 e gzip) para comprimir e empaquetar.

Sintaxe:

```
tar [opcións] arquivo/s
```

- c Crear un arquivo novo e empaquetar
- x Desempaquetar
- t Listar o contido dun arquivo
- r Engadir arquivos ao final dun arquivo
- v Ofrece información dos arquivos que se empaquetan o desempaquetan
- f Especifica o nome do arquivo a ler ou escribir



Fonte: <http://informatica.iessanclemente.net>

Tamén podemos combinar os dous comandos tar e gzip, para crear un arquivo empaquetado y comprimido

```
tar czf fich.tar.gz archivos
```

Para descomprimir y desempaquetar

```
tar xzf fich.tar.gz
```

bzip2

Similar a gzip, pero emprega outro algoritmo de compresión. Bzip2 comprime máis pero consume máis recursos que gzip para comprimir/descomprimir o mesmo arquivo.

- Para comprimir

```
bzip2 fichero
```

- Para descomprimir (2 opcións)

```
bunzip2 fichero.bz2  
bzip2 -d fichero.bz2
```

Como xa vimos, tanto gzip como bzip2 están integrados cocomando tar para que podamos comprimir ou descomprimir varios arquivos ou directorios ao mesmo tempo.

	gzip	bzip2
Comprimir	<code>tar czf fich.tar.gz fich</code>	<code>tar cjf fich.tar.bz2 fich</code>
Descomprimir	<code>tar xzf fich.tar.gz</code>	<code>tar xjf fich.tar.bz2</code>

Z: Indica usar gzip para comprimir
J: Indica usar bzip2 para comprimir

Outros algoritmos de compresión

zip

`zip nomeArquivo.zip nomeArquivo` → Comprime o arquivo
`zip -r nomeArquivo.zip directorio` → Comprime o directorio e todo o seu contido.
`unzip nomeArquivo.zip` → Descomprime

rar

Este paquete está na rama **non-free**

`rar a nomeArquivo.rar f1 f2` → comprime
`unrar x nomeArquivo.rar` → Descomprime

xz

Consigue maiores ratios de compresión en menos tempo. É necesario instalar o paquete **xz-utils**.

`xz nomeArquivo.dat` → comprime e xera nomeArquivo.dat.xz
`xz -d nomeArquivo.dat.xz` → Descomprime

`tar cJf fich.tar.gz fich` → Comprime e empaqueta
`tar xJf fich.tar.gz fich` → Descomprime e desempaqueta

dtrx • "Do The Right Extraction" (Fai unha descompresión correcta)

Permite extraer tar, zip,.. e moitos tipos máis.

Para cada algoritmo de compresión é necesario ter instalado o paquete correspondente.

`dtrx unixmen.tar.gz`

Copias de seguridade en Linux: rdiff-backup

É un script python que nos permite:

- Realizar **copias de seguridade incrementais**.
- Recuperar versións antigas de arquivos.

Vexamos o seguinte exemplo:

Preparativos

- Temos unha carpeta chamada **DATOS**, da que queremos facer copias de seguridade incrementais noutra carpeta chamada **BACKUP**.
- A carpeta DATOS contén:
DATOS/
 - ├─ D2
 - ├─ f1.txt
 - └─ f2.txt
- O arquivo f1.txt contén Línea1

Copias de Seguridade

- Realizamos a copia de seguridade inicial e listamos copias existentes

```
rdiff-backup DATOS/ BACKUP/  
rdiff-backup -l BACKUP/  
Found 0 increments:  
Current mirror: Sun Apr 17 19:47:12 2023
```

- Modifico o arquivo f1.txt e fago outra copia

```
echo Línea2 > DATOS/f1.txt  
rdiff-backup DATOS/ BACKUP/  
rdiff-backup -l BACKUP/  
Found 1 increments:  
increments.2023-04-17T19:47:12+02:00.dir Sun Apr 17 19:47:12 2023  
Current mirror: Sun Apr 17 19:48:04 2023
```

- Borro f1.txt e fago outra copia

```
rm DATOS/f1.txt  
rdiff-backup DATOS/ BACKUP/  
rdiff-backup -l BACKUP/  
Found 2 increments:  
increments.2023-04-17T19:47:12+02:00.dir Sun Apr 17 19:47:12 2023  
increments.2023-04-17T19:48:04+02:00.dir Sun Apr 17 19:48:04 2023  
Current mirror: Sun Apr 17 19:49:50 2023
```

Restaurando versións previas

- Restauramos a primeira copia de seguridade e recuperamos a 1ª versión de f1

```
rdiff-backup -r 2023-04-17T19:47:12+02:00 BACKUP/ Restaurados/  
ls Restaurados/  
f1.txt f2.txt D2
```

- Restauramos a 2ª copia de seguridade e temos a 2ª versión de f2

```
rdiff-backup -r 2023-04-17T19:48:04+02:00 BACKUP/ Restaurados/  
more Restaurados/f1.txt
```

IMPORTANTE:

Cando executamos **rdiff-backup** aparece un aviso indicando que a sintaxe anterior está obsoleta (Deprecated), de momento funciona igual, pero nalgún momento deixará de funcionar.

A nova sintaxe é:

- Copia de Seguridade Incremental

```
rdiff-backup backup ORIXE/ BACKUP/ --print-statistics
```

- Listar os incrementos

```
rdiff-backup list increments BACKUP/
```

- Recuperar unha versión anterior

```
rdiff-backup restore --at 2024-04-04T00:02:12+02:00 BACKUP  
RESTAURADOS/
```

Para máis información: <https://rdiff-backup.net/examples.html>