

## Archivar y comprimir ficheros.

### 1. Empaquetar

#### COMANDO tar

El comando **tar** es utilizado normalmente para archivar o empaquetar ficheros. Su sintaxis general es:

**tar <opciones> <ficheros>**

Las opciones más comunes para *tar* son:

- c** : copia los archivos, los crea.
- v** : muestra mayor información sobre el progreso del comando.
- f** : especifica el archivo destino, pudiendo ser este un archivo de dispositivo.
- u** : agrega uno o más archivos al archivo tar.
- x** : extrae uno o más archivos del archivo tar. Desempaqueta.
- t** : muestra el contenido del archivo tar.

#### Ejemplos :

Supongamos que tenemos el contenido de un directorio.

```
drwxr-xr-x 2 root root 1024 Nov 19 17:57 .
drwxr-xr-x 3 root root 1024 Nov 19 17:54 ..
-rw-r--r-- 1 root root 9660 Nov 19 17:56 a
-rw-r--r-- 1 root root 9660 Nov 19 17:56 b
-rw-r--r-- 1 root root 9660 Nov 19 17:56 c
-rw-r--r-- 1 root root 9660 Nov 19 17:56 d
```

Queremos generar un archivo tar para después hacerle un gzip y mandarlo por mail. Haríamos lo siguiente :

```
[root@linux /adrian]# tar cvf todo.tar *
```

que pasa si se agrega un nuevo archivo al directorio como se ve en el listado :

```
drwxr-xr-x 2 root root 1024 Nov 19 18:04 .
drwxr-xr-x 3 root root 1024 Nov 19 17:54 ..
-rw-r--r-- 1 root root 9660 Nov 19 17:56 a
-rw-r--r-- 1 root root 9660 Nov 19 17:56 b
```

```
-rw-r--r-- 1 root root 9660 Nov 19 17:56 c
-rw-r--r-- 1 root root 9660 Nov 19 17:56 d
-rw-r--r-- 1 root root 28056 Nov 19 18:04 nuevo.txt
-rw-r--r-- 1 root root 51200 Nov 19 18:03 todo.tar
```

si quisiéramos agregarlo, haríamos lo siguiente :

```
[root@linux /adrian]# tar uvf todo.tar nuevo.txt
```

Podemos querer consultar su contenido, con lo siguiente:

```
[root@linux /adrian]# tar tvf todo.tar
```

```
-rw-r--r-- root/root 9660 1999-11-19 17:56 a
-rw-r--r-- root/root 9660 1999-11-19 17:56 b
-rw-r--r-- root/root 9660 1999-11-19 17:56 c
-rw-r--r-- root/root 9660 1999-11-19 17:56 d
-rw-r--r-- root/root 28056 1999-11-19 18:04 nuevo.txt
```

Queremos extraer el archivo "a", hacemos lo siguiente :

```
[root@linux /adrian]# tar xvf todo.tar a
```

Extraemos todos los ficheros

```
[root@linux /adrian]# tar xvf todo.tar
```

## Más ejemplos con tar

### i. Ejemplos de creación de un archivo:

Siendo **docum** un directorio

```
$ tar cf docum.tar docum (empaqueta el directorio docum en el archivo docum.tar)
```

```
$ tar cvf docum.tar docum (muestra los ficheros que se están empaquetando)
```

```
docum/
docum/fichero1
docum/fichero2
docum/fichero3
..... (resto de los ficheros del directorio)
```

```
$ tar cvvf docum.tar docum (al poner varias veces "v", más información durante el proceso)
```

```
drw-r--r-- 1 pepe/curso 0 Jan 30 21:45 docum/
-rw-r--r-- 1 pepe/curso 43 May 3 22:11 docum/fichero1
-rw-r--r-- 1 pepe/curso 72 Jan 13 09:45 docum/fichero2
-rw-r--r-- 1 pepe/curso 88 Sep 23 13:13 docum/fichero3
```

## ii. Ejemplos de extracción de un archivo:

`$ tar xvf docum.tar` (creará un subdirectorio docum y dentro de él todos los archivos desempaquetados con los mismos propietarios y permisos que el original)

```
docum/  
docum/fichero1  
docum/fichero2  
docum/fichero3
```

`$ cd docum`

`$ tar cvf docum.tar *` (empaquetará todos los ficheros que están en el directorio actual pero NO empaqueta el directorio en sí (docum)).

(\*\*\*) Hay que tener en cuenta, a la hora de extraer el contenido de un archivador, si el archivador se creó conservando el nombre del directorio de origen. Es posible que se sobrescriba el contenido de los ficheros originales.

**Ejemplo:** Nos situamos en el directorio raíz como root. Si archivamos los ficheros `/etc/group` y `/etc/passwd`:

```
#tar cvf backup.tar /etc/group /etc/passwd
```

estamos conservando los nombres del directorio al que pertenecen. Por lo tanto, para extraer estos ficheros nos tendremos que situar en el directorio raíz:

```
#cd /  
#tar xvf backup.tar /etc/group /etc/passwd
```

Sin embargo, si archivamos los ficheros `group` y `passwd` estando en `/etc`:

```
#tar cvf /backup.tar group passwd
```

No guardamos la ruta, por lo que para extraer los ficheros tendremos que situarnos en ella:

```
#cd /  
#cd /etc  
#tar xvf /backup.tar group passwd
```

## iii. Extracción de ficheros individualizados:

`$ tar xvf docum.tar docum/fichero2` crearía un subdirectorio docum y en él el fichero2

## 2. Comprimir/ Descomprimir:

### COMANDOS gzip / gunzip

Para comprimir un fichero utilizamos el comando **gzip** y para descomprimirlo **gunzip**.

**gzip**: toma un fichero, lo comprime, salva el fichero comprimido con la extensión **.gz** y borra el fichero original no comprimido.

Su sintaxis: **gzip/gunzip <opciones> <ficheros>**

- c** : Manda el resultado de la compresión al standard output, y sea que deja el archivo origen intacto.
- d** : Se usa para descomprimir.
- f** : Fuerza la compresión o descompresión. No hace ninguna pregunta si desea sobrescribir.
- h** : Muestra la pantalla de ayuda.
- l** : Muestra información sobre los archivos que están comprimidos, no los descomprime.
- n** : Cuando comprime no graba el nombre original del archivo ni la fecha de este. Cuando descomprime no retorna el tamaño original ni la fecha.
- N** : Cuando comprime graba el nombre original y la fecha de este en el archivo gz. Cuando descomprime retorna el nombre original y la fecha del archivo original.
- r** : Comprime recursivamente, o sea comprime el directorio principal y todos sus subdirectorios.
- t** : Chequea la integridad del archivo comprimido.
- v** : Muestra el nombre y el porcentaje de reducción para cada archivo comprimido

El comando **gunzip** tiene los mismos argumentos excepto d.

### Ejemplos:

Si tenemos los siguientes archivos:

|            |    |      |      |      |             |      |
|------------|----|------|------|------|-------------|------|
| drwxr-xr-x | 2  | root | root | 1024 | Dec 6 17:34 | .    |
| drwxr-x--- | 28 | root | root | 2048 | Dec 6 17:34 | ..   |
| -rw-r--r-- | 1  | root | root | 2419 | Dec 6 17:34 | dos  |
| -rw-r--r-- | 1  | root | root | 2419 | Dec 6 17:34 | tres |
| -rw-r--r-- | 1  | root | root | 2419 | Dec 6 17:34 | uno  |

y hacemos un gzip

```
[root@linux /root]# gzip *
```

tendremos:

```
drwxr-xr-x  2 root  root   1024 Dec  6 17:46 .
drwxr-x--- 28 root  root   2048 Dec  6 17:34 ..
-rw-r--r--  1 root  root    52 Dec  6 17:34 dos.gz
-rw-r--r--  1 root  root    52 Dec  6 17:34 tres.gz
-rw-r--r--  1 root  root    52 Dec  6 17:34 uno.gz
```

Para descomprimirlos: [root@linux /root]# **gunzip \***  
y volveremos a tener nuestros archivos originales.

### **Ejemplos:**

1. **\$gunzip backup.tar.gz** (descomprime el fichero backup.tar.gz y obtenemos otro sin .gz).

2. **\$ ls -l prueba**

```
-rw-r--r-- 1 pepe curso 312996 Jan 30 21:44 prueba
```

```
$ gzip prueba
```

```
$ ls -l
```

```
-rw-r--r-- 1 pepe curso 103441 Jan 30 21:45 prueba.gz
```

3. **\$ gzip -l prueba.gz** (da información sobre la compresión realizada)

| compressed | uncompr. | ratio | uncompressed_name |
|------------|----------|-------|-------------------|
| 103441     | 312996   | 67%   | prueba            |

### **3. Combinar tar y gzip/gunzip**

Para archivar un conjunto de ficheros y comprimir el resultado debemos usar en primer lugar el comando tar y a continuación el comando gzip.

Igualmente debemos realizar las operaciones inversas para descomprimir con gunzip, y desempaquetar con tar.

Sin embargo, podemos realizar ambas operaciones en una sola línea de comandos usando el redireccionamiento y tuberías.

Afortunadamente, la orden tar proporciona la opción **z** para comprimir y descomprimir o desempaquetar.

**Ejemplo:**

```
#tar cvfz backup.tar.gz passwd group
#tar xvfz backup.tar.gz
```

**4. Otros compresores y empaquetadores****zip, unzip**

Este comando es una utilidad para empaquetar y comprimir archivos. Para una descripción de los parámetros de zip o unzip ejecútalos sin parámetros en el shell.

Veamos un ejemplo, supongamos que tenemos los siguientes archivos:

```
drwxr-xr-x 2 root root 1024 Nov 16 11:23 .
drwxr-x--- 31 root root 2048 Nov 16 11:16 ..
-rw-r--r-- 1 root root 145200 Nov 16 11:23 caraconmanija
-rw-r--r-- 1 root root 66600 Nov 16 11:19 dedosfinos
-rw-r--r-- 1 root root 60000 Nov 16 11:18 rata
```

Para comprimirlos haríamos lo siguiente : [root@linux /adrian]# **zip todo \***

Lo que generaría un solo archivo llamado todo.zip.

```
Drwxr-xr-x 2 root root 1024 Nov 16 11:26 .
drwxr-x--- 31 root root 2048 Nov 16 11:24 ..
-rw-r--r-- 1 root root 145200 Nov 16 11:23 caraconmanija
-rw-r--r-- 1 root root 66600 Nov 16 11:19 dedosfinos
-rw-r--r-- 1 root root 60000 Nov 16 11:18 rata
-rw-r--r-- 1 root root 1272 Nov 16 11:26 todo.zip
```

Supongamos que llevamos el archivo a otra pc, para descomprimirlos hacemos lo siguiente :

[root@linux /Jaime]# **unzip todo** (Genera los archivos comprimidos antes)

Vamos a ver algunas opciones útiles.

**-r:** comprime todos los subdirectorios y archivos bajo el directorio donde estamos, o donde le indicamos.

**-i:** Incluye solo los archivos que especificamos, por ejemplo:

**# zip todo \* -i \*.txt** (Comprime los archivos que finalicen con .txt de nuestro directorio)

**-f:** Actualiza archivos que ya están comprimidos, basándose en la fecha de creación de estos. Esta opción no funciona como la -u, no agrega archivos nuevos a nuestro zip, solo actualiza los que ya están.

**# zip -f todo** (Actualizará nuestro archivo llamado todo.zip)

**-q:** Trabaja en modo silencioso, elimina todos los mensajes informativos.

**-u:** Actualiza los archivos que hemos modificado, y también agrega archivos a nuestro

.zip.

`#zip -u todo` (funciona como -f agrega las actualizaciones al archivo todo.zip)

`#zip -u todo arch1` (Agrega el archivo arch1 a todo.zip)

-x: Esta opción sirve para excluir uno o varios archivos en nuestro .zip.

`#zip todo * -x \*.Z` (en el ejemplo se excluyen los archivos Z de nuestro .zip).

### **bzip2 /bunzip2 /bzip2recover**

➤ **bzip2** genera archivos de la forma archivo\_original.**bz2**. El comando bzip2 comprime los archivos en bloques de 900kbytes aproximadamente y cada bloque se maneja de manera independiente lo que posibilita que si un bloque es corrupto se puedan recuperar los datos de los bloques que están correctos.

➤ **bunzip2** se utiliza para descomprimir los archivos generados con bzip2.

➤ **bzip2** se puede consultar el contenido de un archivo generado con bzip2.

➤ **bzip2recover** se utiliza para buscar bloques correctos dentro de un archivo bzip2 dañado, el único argumento que necesita este es el nombre del archivo dañado, generando un salida similar a "rec0001fichero.bz2", "rec0002fichero.bz2", etc.

bzip2recover es de mas provecho cuando mas bloques tiene el archivo .bz2, es inútil utilizarlo con archivos de un solo bloque pues este no repara bloques sino extrae los bloques correctos. Si usted desea minimizar la perdida de información por un error del medio físico o de la transmisión debería de achicar el tamaño de los bloques en el momento de la compresión.

### **Ejemplos:**

`tar -c ficheros | bzip2 > archivo.tar.bz2`

Para descomprimir ficheros tar.bz2, se utiliza el siguiente comando:

`bzip2 -dc archivo.tar.bz2 | tar -xv`

Y para ver el contenido de un fichero tar.bz2, se usa el siguiente comando:

`bzip2 -dc archivo.tar.bz2 | tar -t`