



**Universidad Nacional de La Matanza**  
**Ingeniería en Informática-Taller de GNU/Linux**

**TP N° 9**  
**Compresión de Archivos**

**Objetivos:**

- Conocer algunas de las herramientas más utilizadas para el empaquetado y la compresión de archivos.
- Aprender la utilización básica de estas herramientas.

**Guía:**

**Comandos utilizados para la compresión y descompresión de archivos**

**Comando tar**

El comando tar es utilizado normalmente para empaquetar archivos. El comando tar no comprime automáticamente los archivos mientras los empaqueta. El formato del comando tar es:

```
tar <opciones> <archivo1> <archivo2> ...<archivoN>
```

donde <opciones> es la lista de comandos y opciones para tar, y <archivo1> hasta <archivoN> es la lista de archivos a añadir o extraer del archivo empaquetado. Por ejemplo, el comando

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

empaquetará todos los archivos de /etc en el archivo backup.tar.

En el primer argumento de tar ("cvf"), la opción "c" le dice a tar que cree un nuevo archivo (create). La opción "v" fuerza a tar en el modo verbose, imprimiendo los nombres de los archivos según se empaquetan. La opción "f" le dice a tar que el siguiente argumento (backup.tar) es el nombre del archivo a crear. El resto de los argumentos de tar son los nombres de archivos y directorios a añadir al archivo empaquetado.

**Descomprimir archivos con tar**

```
# tar xvf backup.tar
```

Esto, extraerá el archivo backup.tar en el directorio actual. Esto puede ser peligroso, porque si el archivo ya existía se sobrescribirá. Por otra parte, antes de extraer archivos tar es importante conocer donde se deben desempaquetar.

Por ejemplo, digamos que se empaquetaron los siguientes archivos: /etc/hosts, /etc/group, y /etc/passwd. Si se usó:

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

el nombre de directorio /etc se añadió al principio de cada nombre de archivo. Para poder extraer los archivos en el directorio correcto, se necesitará utilizar los siguientes comandos:

```
# cd /  
# tar xvf backup.tar
```

Ya que los archivos se extraen con el nombre de camino almacenado. Sin embargo, si se empaquetaron los archivos con los comandos:

```
# cd /etc  
# tar cvf hosts group passwd
```

Los nombres de directorio no se salvaron en el archivo empaquetado. Por esto se necesitara hacer "cd /etc" antes de extraer los archivos. Como se puede ver, el cómo haya sido creado un archivo tar marca una gran diferencia en como se extrae. Se puede usar el comando:

```
# tar tvf backup.tar
```

para mostrar un "índice" del archivo tar antes de desempaquetarlo. De esta forma se puede ver que directorio se utilizó como origen de los nombres de los archivos, y se puede extraer el archivo desde la localización correcta.

***Además agregando el switch 'z' utiliza el gzip y el gunzip para comprimir / descomprimir.***

## **Comando gzip**

El comando gzip se utiliza para comprimir un archivo. El formato del comando gzip es:

```
gzip archivo  
(archivo se convierte en archivo.gz comprimido)  
  
gzip -c archivo > archivo.gz  
(Se crea archivo.gz, archivo no se modifica)
```

Por ejemplo:

```
# gzip -9 backup.tar
```

Comprimirá backup.tar y lo dejará como backup.tar.gz, que es la versión comprimida del archivo tar. La opción -9 le dice a gzip que utilice el mayor factor de compresión.

## **Descomprimir archivos con gunzip o con gzip**

El comando gunzip se utiliza para descomprimir un archivo comprimido con gzip. Equivalentemente, se puede utilizar "gzip -d".

El formato del comando gunzip es:

```
gunzip archivo.gz (descomprime archivo.gz)  
  
gunzip -c archivo.gz > nuevo  
(descomprime archivo.gz y se redirecciona la salida al archivo nuevo).
```

## **Comprimir y empaquetar**

Por lo tanto, para empaquetar un grupo de archivos y comprimir el resultado, se pueden utilizar los comandos:

```
# tar cvf backup.tar /etc
# gzip -9 backup.tar
```

El resultado será backup.tar.gz.

Para desempaquetar este archivo, se usan los comandos contrarios:

```
# gunzip backup.tar.gz
# tar xvf backup.tar
```

### **Comando zcat**

Es similar al cat, pero pasando el archivo a través del gunzip. Con zcat podemos, ver el contenido de un archivo comprimido sin descomprimirlo, similar al cat. El formato del comando zcat es:

```
# zcat <archivo-de-texto-comprimido>
```

### **Comandos zip y unzip**

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

El formato del comando zip es:

```
# zip <nombre_archivo_comprimido> <archivos_a_comprimir>
```

Si se quieren comprimir todos los archivos de un directorio se debe usar \* en <archivos\_a\_comprimir>.

Para descomprimirlos se debe usar el comando unzip. El formato del comando unzip es:

```
# unzip <nombre_archivo_comprimido>
```

Algunas opciones útiles de estos comandos:

**-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 (solo comprimiría los archivos que finalicen con .txt).
```

**-f:** Si ya hemos creado un .zip de nuestros archivos, pero periódicamente modificamos estos, con este argumento no hace falta crear el .zip de nuevo, este agrega los cambios de nuestros archivos en el zip, basándose en la fecha de creación de estos. Esta opción no funciona como -u, no agrega archivos nuevos a nuestro zip, solo actualiza los que ya están.

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

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

```
# zip -u <nombre_archivo_comprimido> <archivo_a_agregar>
```

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

```
# zip <nombre_archivo> * -x \*.Z (se excluyen los archivos Z de nuestro .zip)
```

**-y** : Esta opción sirve para guardar los links en el zip como links en vez de guardar el archivo al que apunta el link.

## **Comandos bzip2/bunzip2/bzcat/bzip2recover**

El comando bzip2 comprime archivos usando el algoritmo Burrows–Wheeler block–sorting text compression y Huffman coding. La línea de comandos es similar al GNU gzip pero no es idéntica. Tiene un poco más de capacidad de compresión que otros compresores más convencionales basados en LZ77/LZ78. Este genera archivos de la forma archivo\_original.bz2.

El comando bunzip2 se utiliza para descomprimir los archivos generados con bzip2.

Con bzcat se puede consultar el contenido de un archivo de texto comprimido con bzip2.

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 está corrupto se puedan recuperar los datos de los bloques que están correctos. El programa 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 más provecho cuanto más bloques tiene el archivo bz2. Es inútil utilizarlo con archivos de un solo bloque pues este no repara bloques sino que extrae los bloques correctos. Si se desea minimizar la pérdida de información por un error del medio físico o de la transmisión se debería achicar el tamaño de los bloques en el momento de la compresión.

## **Comando cpio**

Si bien cpio no sirve para comprimir, nos ayudará a empaquetar todos nuestros archivos en uno solo para después comprimirlo con alguna utilidad. Con cpio se puede empaquetar archivos y mandarlos al estándar output, a un archivo, o a un dispositivo.

El comando cpio tiene tres modos de operación.

El modo copia–afuera, cpio copia varios archivos de entrada hacia otro único de salida. Este modo funciona leyendo una lista de archivos, uno por línea de la entrada estándar, escribiendo el archivo en la salida estándar, este modo se representa utilizando el argumento **-o**.

El modo copia–adentro cpio copia archivos desde un archivo de entrada. Lee el archivo desde la entrada estándar, este modo se representa utilizando el argumento **-i**.

El modo copia–de–paso cpio copia archivos desde un árbol de directorio a otro, sin utilizar utilizar ningún archivo, este modo se representa utilizando el argumento **-p**.

Nota: El comando ls genera la lista que es mandada a través de un pipe para que cpio la reciba, y empaquete los archivos.

Cuando queremos sacar los archivos ejecutamos el siguiente comando:

```
# cpio -iv < archivo.cpio
```

Si ya hubiese archivos con el mismo nombre y quisiéramos sobrescribirlos utilizaríamos el argumento u.

```
# cpio -iuv < archivo.cpio
```

Para ver solamente el contenido del archivo cpio el comando será:

```
# cpio -tv < archivo.cpio
```

## Comando unrar

El comando unrar permite descomprimir archivos comprimidos con la aplicación rar. El mecanismo de compresión del rar es propietario, razón por la cual no existen compresores tipo rar de libre uso.

*El texto anterior es una adaptación de:*

<http://lucas.hispalinux.es/Manuales-LuCAS/LIPP/lipp-1.1-html-2/lipp.htm>

---

## Ejercicios

1) Buscar algún archivo de texto comprimido con gzip (\*.gz);

Ayuda: para buscar archivos, utilizar el comando "locate"

Archivo elegido:

2) Utilizar cat para ver el contenido. (Si la terminal queda mal configurada, comando "reset")


3) Utilizar zcat para ver el contenido. Si es muy largo el texto pasarlo a través de less o more.


4) descomprimir el archivo ( en /tmp)

--

5) Verificar que esté sin compresión y comprimirlo nuevamente en el mismo formato .


6) Hacer un .tar.gz de todos los archivos en /bin. Utilizar los comandos tar y gzip unidos con un pipe.


7)descomprimirlos en /tmp

8) Obtener una salida equivalente a la que se obtendría realizando:

```
# tar cvf backup.tar /etc
# gzip backup.tar
```

pero en una sólo línea y utilizando sólo el comando tar.

9) comprimir todos los archivos de /etc en un archivo llamado backup\_etc utilizando el comando zip. Descomprimirlo en ~/etc usando el comando unzip

  

10)Generar un archivo cpio que contenga todos los archivos del directorio /sbin.

11) Extraer los archivos desde el archivo cpio creado en el punto anterior en el directorio ~/sbin.

12) Generar un archivo bzip2 que contenga todos los archivos del directorio /sbin.

Generar un archivo zip que contenga todos los archivos del directorio /sbin.

Generar un archivo tar.gz que contenga todos los archivos del directorio /sbin.

Anotar los tamaños de los archivos resultantes

**Nota:** Si los tamaños difieren demasiado, puede deberse a la inclusión o no de "links" en la compresión. Para que la comparación sea válida, incluir los archivos apuntados con links dentro de los archivos comprimidos. (El significado de links o enlaces se verá mas adelante en el curso)

Sbin.cpio	Sbin.zip
Sbin.bz2	Sbin.tar.gz