

```
$ tar -cf practicas.tar practicas
$ gzip practicas.tar
$ ls *.gz
practicas.tar.gz
```

Para descomprimir se utiliza `gunzip(1)`, y se cambia el orden de los comandos:

```
$ gunzip practicas.tar.gz
$ tar -xf practicas.tar
```

Este par de operaciones es tan común que `tar` permite añadir la opción `-z` para trabajar directamente sobre paquetes comprimidos. Las dos secuencias de comandos anteriores se pueden simplificar como sigue:

```
$ tar -cfz practicas.tar.gz practicas
$ tar -xfz practicas.tar.gz
```

4. Cómo obtener ayuda

Utilidades empleadas

man, info, apropos, whatis, pinfo

Comando	Función
<code>man(1)</code>	Ver páginas de manual
<code>info(1)</code>	Ver páginas <i>info</i> , normalmente más largas que las de manual
<code>pinfo(1)</code>	Similar a <code>info(1)</code> , pero más bonito y fácil de utilizar
<code>apropos(1)</code>	Busca un texto en los resúmenes de las páginas de manual
<code>whatis(1)</code>	Muestra el resumen de la página de manual para un comando dado.

Algunas opciones interesantes de `man(1)`:

```
-s          Sección a utilizar para buscar la página de manual.
-a          Ver manuales de todas las secciones.
-k          Adoptar el funcionamiento de apropos(1).
-f          Adoptar el funcionamiento de whatis(1).
```

Las dos últimas son especialmente útiles si algún sistema no dispone de las correspondientes herramientas.

Unix y el shell Bash

Adrián Pérez — moebius@gpul.org

Según el diccionario Merriam-Webster:

Shell *Any of the spaces occupied by the orbits of a group of electrons of approximately equal energy surrounding the nucleus of an atom.*

Bash *To attack physically or verbally.*

1. Sistema de ficheros: lo básico

Utilidades empleadas

ls, find, mkdir, rmdir, rm, cp, mv, touch

Con `ls(1)` se pueden listar los contenidos de un directorio, formateando el listado con la casi ilimitada potencia de las opciones del comando (la mayoría de la gente sólo conoce `-l`, `-h` y `-a`):

```
-a, --all          Muestra todos los ficheros, incluso los ocultos.
-l, --format=long  Produce un listado detallado, incluyendo entre otros fechas, tamaño, propietario, grupo y permisos.
-h, --human-readable  Imprime los tamaños de los elementos de forma más legible.
```

```
$ ls /
bin  dev  home  lost+found  mnt  proc  sbin  tmp  var
boot  etc  lib  media      opt  root  sys  usr
$ ls -lh /
total 68K
drwxr-xr-x  2 root   root  4.0K 2006-10-22 16:55 bin
drwxr-xr-x  4 moebius users  77 2006-10-24 23:47 media
lrwxrwxrwx  1 root   root   5 2006-04-11 03:11 mnt -> media
drwx----- 49 root   root  4.0K 2006-10-24 20:13 root
...
```

Las combinaciones de todas las opciones pueden llegar a ser simpáticas:

```
$ ls -lisa
$ ls -bart
$ ls -simpson
```

Para crear y eliminar directorios se utilizan `mkdir(1)` y `rmdir(1)`. Los directorios deben estar vacíos para poder ser borrados:

```

$ mkdir directorio
$ ls -l
total 0
drwx----- 2 moebius users 6 2006-10-24 23:59 directorio
$ rmdir directorio
$ ls -l
total 0

```

Es posible crear un fichero vacío con el comando `touch(1)`. Puede parecer una tontería, pero este comando puede ser tan útil como para crear ficheros para los ejemplos del taller¹:

```

$ touch a b c ; ls
a b c

```

Para eliminar, copiar y renombrar ficheros y directorios se utilizan `rm(1)`, `cp(1)` y `mv(1)`²:

```

$ mkdir arradio
$ mv arradio radio
$ touch fm am mw
$ cp fm am mw radio
$ rm radio/mw
$ rm radio
rm: cannot remove 'radio': Is a directory

```

Para que los comandos operen sobre un árbol de directorios completo muchos de ellos admiten una opción `-R` (o `-r` en algunos casos):

```

$ rm -r radio

```

2. Buscando agujas en el pajar

Utilidades empleadas

`pwd`, `find`, `who`, `whoami`, `hostname`, `whatis`

Esta serie de comandos son, en general, informativos³:

Comando	Función
<code>pwd</code>	Directorio de trabajo
<code>whoami</code>	Nombre de usuario en uso
<code>who</code>	Usuarios conectados
<code>hostname</code>	Nombre de la máquina
<code>whatis</code>	Descripción de un comando
<code>find</code>	Busca ficheros usando diversos criterios

Algunos ejemplos:

¹De hecho además de crear un fichero vacío si no existiese, también cambia la fecha de modificación de los ya existentes, lo cuál a su vez también puede parecer inútil... si existen es que son de uso común.

²Por la brevedad de los nombres podemos deducir que los autores de Unix eran muy vagos.

³Particularmente `find(1)` es tan extenso que requeriría un taller dedicado a él para comprender una porción de todo lo que puede hacer.

```

$ whatis pwd
pwd                (1p) - return working directory name
pwd.h [pwd]       (0p) - password structure
$ hostname
tron
$ pwd
/home/moebius
$ whoami
moebius
$ who loves you
moebius pts/4      2006-10-24 12:11 (:0.0)
$ find ~ -name 'ee*' ( -type f -o -type d )
/home/moebius/peer.txt
/home/moebius/fleet/

```

Para el caso de `find(1)` se presentan algunas de las opciones más comunes:

- name, -iname** Patrón que debe seguir el nombre, en el segundo caso no se distinguen mayúsculas y minúsculas.
- type** Tipo de fichero: `f` para archivos normales, `d` para directorios.
- size** Tamaño del fichero.
- not** Negación lógica.
- o** Disyunción lógica (*or*).
- delete** Borra los ficheros encontrados.
- maxdepth** Profundidad máxima de directorios.

3. Examinar y tratar ficheros

Utilidades empleadas

`more`, `less`, `cat`, `split`, `tar`, `gzip`

Tanto `more(1)` y `less(1)` pueden ser empleados para ver ficheros de texto. Normalmente se utiliza el segundo, ya que `more(1)` no permite volver hacia atrás en el texto — aunque se comenta aquí puesto que en sistemas antiguos puede no existir `less(1)`. Aunque no es su propósito original, `cat(1)` también es útil para obtener el contenido de ficheros:

```

$ cat /usr/share/*/COPYING
$ less /usr/share/*/COPYING

```

La función principal de `cat(1)` es concatenar ficheros — por ejemplo para reconstruir un fichero grande que ha sido separado en pedazos con `split(1)`:

```

$ split -b 100k /bin/bash trozo_bash_
$ ls trozo_bash_*
trozo_bash_aa trozo_bash_ac trozo_bash_ae trozo_bash_ag
trozo_bash_ab trozo_bash_ad trozo_bash_af
$ cat trozo_bash_* > bash_reconstruido

```

Para comprimir ficheros y directorios en un único fichero se utiliza el programa `tar(1)`, y `gzip(1)` para comprimir: