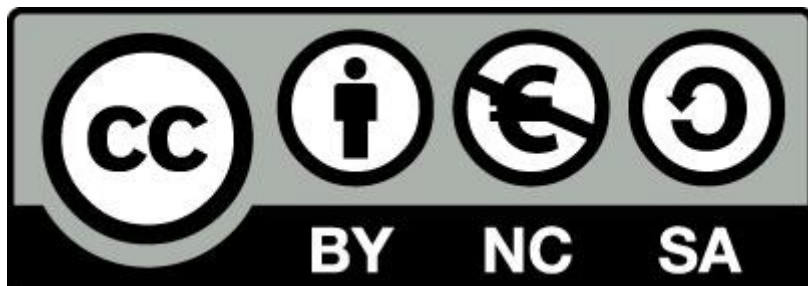


MANUAL BÁSICO DE COMANDOS LINUX
SISTEMAS OPERATIVOS MONOPUESTO

Mario García Alcázar

Esta obra esta sujeta a la Licencia Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/3.0/es/> o envíe una carta Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.



Última revisión Enero de 2016.

Índice de contenido

1. Introducción.....	6
2. Formato de los comandos.....	7
3. Comandos Linux.....	8
3.1 Comandos generales.....	8
Comando clear.....	8
Comando cal.....	8
Comando date.....	8
Comando echo.....	9
Comando expr.....	9
Comando history.....	10
Comando hostname.....	10
Comando finger.....	10
Comando man.....	10
Comando reboot.....	10
Comando set.....	11
Comando startx.....	11
Comando shutdown.....	11
Comando time.....	11
Comando uptime.....	11
Comando uname.....	12
Comando who.....	12
Comando whoami.....	12
2.2 Comandos de gestión de ficheros.....	13
Comando cat.....	13
Comando cd.....	13
Comando chmod.....	13
Comando chown.....	16
Comando chgrp.....	16
Comando cmp.....	16
Comando cp.....	16
Comando cut.....	17
Comando diff.....	17
Comando find.....	18
Comando file.....	18
Comando grep.....	18
Comando head.....	19
Comando ln.....	19
Comando ls.....	19
Comando more.....	20
Comando mv.....	20
Comando mkdir.....	20
Comando nl.....	20
Comando pwd.....	20

Comando rm.....	21
Comando rmdir.....	21
Comando sort.....	21
Comando tail.....	21
Comando touch.....	22
Comando tr.....	22
Comando uniq.....	22
Comando wc.....	23
Comando whereis.....	23
2.3 Comandos de gestión de procesos.....	24
Comando at.....	24
Comando atq.....	24
Comando atrm.....	24
Comando bg.....	25
Comando crontab.....	25
Comando free.....	26
Comando kill.....	27
Comando lsof.....	28
Comando ps.....	29
Comando pstree.....	30
Comando sleep.....	30
Comando top.....	31
Comando watch.....	31
2.4 Comandos de gestión de usuarios.....	32
Comando addgroup.....	32
Comando adduser.....	32
Comando chfn.....	33
Comando chsh.....	33
Comando delgroup.....	34
Comando deluser.....	34
Comando groups.....	35
Comando passwd.....	35
Comando su.....	35
Comando umask.....	35
2.5 Comandos de empaquetado y compresión.....	37
Comando add-apt-repository.....	37
Comando apt-get.....	37
Comando apt-cache.....	38
Comando pgkg.....	38
Comando gdebi.....	39
Comando tar.....	39
Comando gpg.....	40
Comando gpg-zip.....	40
Comando gzip.....	41
2.6 Comandos de gestión de discos y dispositivos.....	42
Comando df.....	42

Comando dmesg.....	42
Comando du.....	42
Comando fdisk.....	42
Comando e2fsck.....	44
Comando lshw.....	44
Comando lspci.....	44
Comando mkfs.....	44
Comando mkswap.....	45
Comando swapoff.....	45
Comando swapon.....	46
Comando mount.....	46
Comando umount.....	48
2.7 Comandos de gestión de red.....	50
Comando ifconfig.....	50
Comando ifdown.....	51
Comando ifup.....	51
Comando ping.....	51

1. Introducción.

Este manual tiene como objetivo especificar el funcionamiento de los principales comandos de consola que podemos necesitar para la explotación o administración de sistemas Linux basados en la distribución **DEBIAN**.

La razón de centrarnos en **DEBIAN** es su amplia aceptación tanto para la administración de servidores como para su uso en ordenadores PC ya sea con dicha distribución o con versiones basadas en ella como Ubuntu, Linux Mint, etc.

Por otra parte, hay que indicar, que el objetivo de este manual no es especificar cada una de las opciones de dichos comandos, sino su funcionamiento general y las opciones más comunes que puedan ser de utilidad. Para más información sobre cada comando, será necesario referirse a la ayuda de Linux, por medio del comando **man**.

En este manual, se han recogido los comandos agrupados según diferentes conjuntos de funcionalidades para que sean más fáciles de localizar.

2. Formato de los comandos.

Los comandos Linux que introducimos en una consola shell, tienen la siguiente sintaxis:

COMANDO [ARGUMENTOS] [PARÁMETROS]

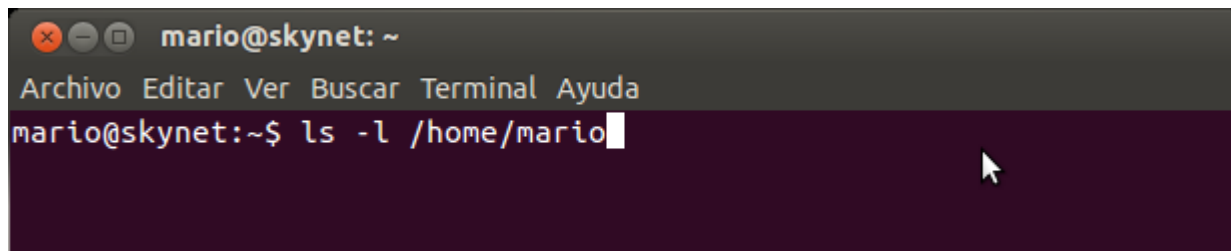
Los elementos que aparecen entre corchetes son **optativos**, y deberemos ponerlos según el tipo de comando y las opciones con las que queramos que se ejecute.

El **COMANDO** es el nombre de la orden a ejecutar (ls, ps cat, etc).

Los **ARGUMENTOS** son las opciones de dicho comando y varían según el tipo de comando concreto.

Los **PARÁMETROS** indican valores necesarios para la correcta ejecución del comando.

Por ejemplo, en el caso de la consulta del contenido de un directorio, la orden completa seria la siguiente.

A screenshot of a Linux terminal window. The title bar shows window control buttons and the text 'mario@skynet: ~'. Below the title bar is a menu bar with the options 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal prompt is 'mario@skynet:~\$' and the command 'ls -l /home/mario' is being entered, with a cursor at the end of the line.

En este caso, el comando seria **ls**, los parámetros “**-l**” y los argumentos serian el directorio que se quiere consultar “**/home/mario**”.

3. Comandos Linux.

Seguidamente se describen los principales comandos Linux, referidos a los siguientes grupos de funcionalidades: generales, gestión de ficheros, gestión de usuarios, gestión de procesos, compresión y gestión de red.

3.1 Comandos generales.

En este apartado, introduciremos comandos generales del sistema que pueden ser de utilidad.

Comando clear

Limpia la pantalla

Formato: clear

Comando cal

Muestra una cadena por la salida estándar con el calendario. Si no se especifican opciones muestra el mes actual.

Formato: cal [OPCIONES]

Opciones básicas:

-3	muestra el mes actual el anterior y el siguiente
mes año	muestra el calendario del mes del año indicado
año	muestra el calendario de todos los meses del año indicado.

Comando date

Muestra una cadena con la fecha y hora del sistema por la salida estándar.

Formato: date [OPCIONES] [[mes] año]

Opciones básicas:

-s “cadena_de_fecha”	Permite cambiar la fecha y hora del sistema. El formato de la cadena de fecha es: mm/dd/yyyy hh:mm Ej.- date -s “12/30/2000 8:00”. Debemos de ser root para poder hacer esto. No obstante, hay que indicar que Linux al arrancar coge otra vez la fecha de la BIOS.
----------------------	---

Comando echo

Muestra una cadena de caracteres por la salida estándar.

Formato: echo [CADENA]

Opciones básicas:

-e	Permite introducir tabulaciones o saltos de página. Ejemplo: echo -e "Hola /n/n/n"
----	--

Notas:

La cadena puede expresarse de diferentes formas:

- Con comilla invertida ` `.- Se ejecuta y el resultado es lo que se imprime.
Ej.- echo `ls -l`--> muestra el resultado del comando ls -l
- Con comilla normal ' '.- Se toma como un texto fijo. No se interpretarían las variables de entorno.
Ej.- A=1
echo '\$A' ---> \$A
- Sin comillas o con comilla doble.- Se interpretan las variables de entorno y se muestra el resultado.
Ej.- A=1
echo "hola \$A" ---->hola 1

Comando expr

Permite ejecutar operaciones aritméticas. Mostrando el resultado por la salida estándar.

Formato: expr [OPERACIÓN ARITMÉTICA]

Los operadores que podemos usar son los siguientes:

OPERADOR	USO
+	suma
-	resta
*	multiplicación
/	división
%	resto

Ejemplos:

expr 1 + 2 → 3

expr \$A * 3 → Multiplica 3 por el contenido de la variable de entorno A

expr 5 % 2 → Muestra el resto de dividir 5 entre 2

expr `expr 3 + 2` / 5 → Implementación de paréntesis (3+2)/5 = 1

Comando history

Muestra los últimos comandos ejecutados por el usuario

Formato: history

Comando hostname

Muestra una cadena por la salida estándar con el nombre del equipo

Formato: hostname

Comando finger

Muestra una cadena por la salida estándar con información detallada de todos los usuarios conectados en el sistema. Si se especifica un nombre de usuario, solo mostrará información de este.

Formato: finger [usuario]

Comando man

Muestra información de ayuda sobre un comando Linux.

Una vez consultada la ayuda, volvemos al prompt del sistema con la tecla **q**

Formato: man comando_linux

Comando reboot

Permite reiniciar el sistema de forma controlada. Para ejecutarlo es necesario tener permisos de administrador.

Formato: reboot

Comando set

Muestra la lista completa de variables de entorno por la salida estándar.

Formato: set

Comando startx

Inicia el modo gráfico, cuando se arrancó el sistema en modo texto.

Formato: startx

Comando shutdown

Permite apagar el equipo de forma controlada. Para ejecutarlo es necesario tener permisos de administrador.

Formato: shutdown

Comando time

Muestra el tiempo que tarda en ejecutarse una orden cualquiera. El texto que devuelve el comando tiene tres partes:

- **real** indica el tiempo efectivo de ejecución de dicho comando.
- **user** define el tiempo de CPU utilizado.
- **sys** indica el tiempo empleado en realizar llamadas al sistema.

Formato: time COMANDO

Ejemplo: time ls -l

Comando uptime

Muestra el tiempo que lleva el sistema activo. Concretamente muestra: la hora actual, el tiempo de funcionamiento del sistema, usuarios conectados, la carga media del sistema durante el último minuto, los últimos 5 minutos y los últimos 15 minutos.

Para ver además de esta información más detallada de los usuarios conectados al sistema (como uso de CPU) y comando actual, podemos usar el comando **w**

Formato: uptime

Comando uname

Muestra información sobre el sistema operativo instalado.

Formato: uname [OPCIONES]

Opciones básicas:

-s	Muestra el nombre del kernel
-o	Muestra el sistema operativo.
-a	Muestra todos los datos. los más importantes son: sistema operativo, red, hostname, nombre del kernel, versión del kernel y procesador del sistema.

Comando who

Muestra una cadena por la salida estándar con la lista de usuarios conectados al sistema.

Formato: who

Opciones básicas:

-r	Indica el runlevel en el que estamos operando.
-b	Muestra el momento del último arranque del sistema.
-all	Indica información detallada de los usuarios conectados. Incluye la mayoría de las opciones del comando

Comando whoami

Muestra una cadena por la salida estándar con el nombre del usuario actual con el que estamos trabajando.

Formato: whoami

2.2 Comandos de gestión de ficheros.

En este apartado se describen los comandos que nos permiten trabajar con ficheros, modificarlos y acceder a su contenido.

Comando cat

Muestra el contenido de un fichero.

Formato: cat NOMBRE_DE_FICHERO

Comando cd

Cambia el directorio actual por el especificado por parámetro. Este parámetro puede ser tanto una ruta absoluta como una ruta relativa.

Formato: cd RUTA

Comando chmod

Modifica los permisos de un fichero o directorio. Para poder modificar los permisos de un fichero hemos de ser root o el usuario propietario de dicho fichero. Cualquier otro usuario no podrá modificar dichos permisos, aunque tuviese permisos de escritura.

Formato: chmod [OPCIONES] PERMISOS[,PERMISOS] FICHERO

Opciones básicas:

-R	Opera en directorios de forma recursiva.
----	--

Permisos:

Hay dos modalidades, por medio de un **número octal** o utilizando **codificación con caracteres**:

- En el caso de la **codificación caracteres**, especificaremos los permisos con una o mas secuencias construidas según el siguiente formato y separadas mediante comas:
 - Usuarios sobre los que se actúa:
 - u -> usuario propietario.
 - g -> grupo propietario.
 - o -> resto de usuarios.

- a -> todos los usuarios (por defecto si no ponemos nada se usa este)
- Se indica el símbolo + o – según queramos poner o quitar permisos.
- Se indican los permisos a poner o quitar:
 - r -> lectura.
 - w -> escritura.
 - x -> ejecución
 - Atributos especiales:
 - s: los atributos suid y sgid, otorgan a un "fichero" los permisos de su dueño o grupo respectivamente, cada vez que se ejecute, sea quien sea el que lo ejecute. El atributo s nunca puede aplicarse al resto de usuarios, solo al propietario o al grupo. Además se muestra en un ls -l con la sigla 'S'
 - Ejemplo: `chmod g+s /usr/bin/cdrecord` activa el bit sgid en `cdrecord`
 - t: el atributo sticky (pegajoso) hace que sólo el propietario del fichero pueda borrarlo.
 - Ejemplo: `chmod o+t fichero` activa el atributo pegajoso. Es equivalente a `chmod +t` y se muestra con un ls -l con la sigla 'T'

Ejemplo.-

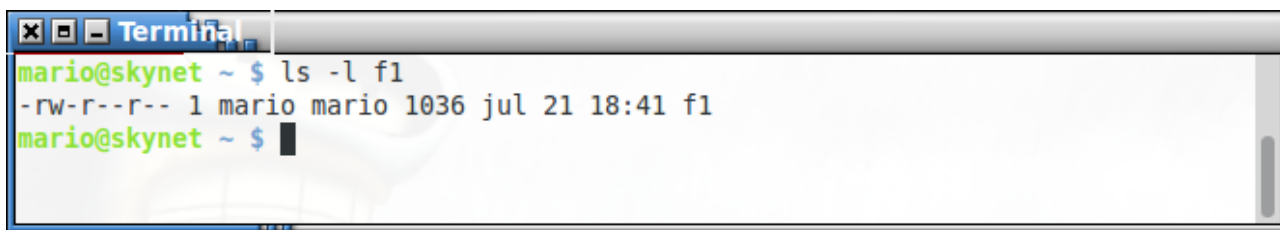
`chmod u+x,g-rw f1` → Este comando da permisos de ejecución al usuario y quita los permisos de lectura y escritura al grupo propietario.

- Con el sistema del **número octal**:
 - Se especifican los permisos creando una máscara en binario donde se pone 1 en la posición de cada permiso que se quiera añadir y 0 en los que se deseen quitar.
 - Posteriormente se convierte dicha máscara a octal.

Ejemplo.- `chmod 600 fich1` y daría permisos de lectura y escritura al propietario del fichero y quitaría cualquier tipo de permiso al grupo propietario y al resto de usuarios.

Notas: Recordamos en este punto el formato de los permisos en ficheros Linux:

Si ejecutamos el comando `ls -l` sobre un fichero obtenemos diversos datos, el primero de los cuales indica los permisos de dicho fichero.



```
mario@skynet ~ $ ls -l f1
-rw-r--r-- 1 mario mario 1036 jul 21 18:41 f1
mario@skynet ~ $
```

En este caso los permisos del fichero f1 serían: **-rw-r--r--**

La interpretación de estos símbolos es la siguiente:

- El primer carácter indica el **tipo de archivo**:
 - **_** Archivo normal.
 - **d** Directorio.
 - **l** Enlace simbólico.
 - **c** Archivo de dispositivo tipo carácter.
 - **b** Archivo de dispositivo tipo bloque.
- Los siguientes tres caracteres indican los permisos que tiene sobre el fichero **su usuario propietario**.
 - Si tiene una **r** en la primera posición es que puede leerlo.
 - Si tiene una **w** en la segunda posición es que puede modificarlo.
 - Si tiene un **x** en la tercera posición es que puede ejecutarlo.
- Los siguientes tres caracteres indican los permisos que tiene sobre el fichero **los usuarios de su grupo propietario**.
 - Si tiene una **r** en la primera posición es que puede leerlo.
 - Si tiene una **w** en la segunda posición es que puede modificarlo.
 - Si tiene un **x** en la tercera posición es que puede ejecutarlo.
- Los siguientes tres caracteres indican los permisos que tiene sobre el fichero **el resto de usuarios del sistema**.
 - Si tiene una **r** en la primera posición es que puede leerlo.
 - Si tiene una **w** en la segunda posición es que puede modificarlo.
 - Si tiene un **x** en la tercera posición es que puede ejecutarlo.
- Existen también dos atributos especiales:
 - **S**: los atributos **suid** y **sgid**, otorgan a un "fichero" los permisos de su dueño o grupo respectivamente, cada vez que se ejecute, sea quien sea el que lo ejecute. El atributo **s** nunca puede aplicarse al resto de usuarios, solo al propietario o al grupo. Este atributo aparece en la posición de la 'x' de los permisos de usuario (**suid**) o en la

- posición de la 'x' de los permisos del grupo (guid).
- **T**: el atributo sticky (pegajoso) hace que sólo el propietario del fichero pueda borrarlo. Este atributo aparece solamente en la posición de la 'x' de los permisos del reto de usuarios.

Comando chown

Cambia el usuario propietario y/o el grupo del fichero o directorio.

Esta operación solo puede hacerla el usuario root. Esto es así ya que en caso contrario, un usuario podría elegir un script de root o de otro usuario, ponerse como propietario y acto seguido modificar sus permisos y ejecutarlo.

Formato: chown [OPCIONES] PROPIETARIO[:GRUPO] FICHERO

Opciones básicas:

-R	Opera en directorios de forma recursiva.
----	--

Comando chgrp

Cambia el grupo propietario de un fichero.

Al igual que el coman chown, esta operación solo puede realizarla el usuario root.

Formato: chgrp [OPCIONES] GRUPO FICHERO

Opciones básicas:

-R	Opera en ficheros y directorios de forma recursiva.
----	---

Comando cmp

Compara dos ficheros byte a byte y muestra la primera diferencia entre ellos.

Formato: cmp [OPCIONES] FICHERO1 [FICHERO2]

Opciones básicas:

-l	Muestra todas las diferencias y no solo la primera.
----	---

Comando cp

Copia un fichero o directorio. Hay que introducir el fichero o ficheros origen y un directorio destino de la copia.

Manual básico de comandos Linux DEBIAN

Formato: cp [OPCIONES] fichero1 [fichero2 .. ficheroN] destino

Opciones básicas:

-R	Copia un directorio recursivamente
-p	Copia preservando permisos, propietario, grupos y fechas.
-d	Conserva los enlaces simbólicos.
-a	Equivalente a -dpR

Comando cut

Permite dada una cadena de entrada, obtener un conjunto de caracteres de cada una de sus líneas de texto

Formato: cut [OPCIONES] [FICHERO]

Opciones básicas:

-c	Obtiene caracteres sueltos.
-d	Establece un delimitador para la selección por columnas.
-f	Indica las columnas que se quieren obtener usando el delimitado especificado con la opción -d

Ejemplos:

cut -c1-5 → Obtiene los 5 primeros caracteres de cada línea del texto de entrada.

cut -d ':' -f1,3 → Obtiene los textos de las columnas 1 y 3 de cada línea del texto de entrada, tomando como separador de columna el carácter ':'

Comando diff

Compara ficheros o el contenido de directorios. Es más avanzado que cmp.

Formato: diff [OPCIONES] FICHERO1 [FICHERO2]

Opciones básicas:

-b	Ignora los espacios en blanco
-text	Compara el texto línea por línea.
-w	Descarta espacio en blanco cuando compara líneas.
-r	Compara directorios de forma recursiva.

-q	Informa sólo de si los ficheros difieren.
-R	Muestra la salida a dos columnas.

Comando find

Busca ficheros en la estructura de directorios, que cumplan una serie de condiciones. Si no se especifica la ruta de búsqueda, empezará desde el directorio actual.

Formato: find [RUTA] [OPCIONES]

Opciones básicas:

-name CADENA	Busca en fichero con el nombre especificado. Permite usar comodines. Para evitar errores es conveniente poner el nombre entre ' ' en el caso de usar comodines. Ej.- -name '*t*' (Busca archivos que tengan una letra t)
-perm /PERMISOS	Busca los archivos con los permisos especificados. Ej.- -perm /777 o -perm /u+w,o+x
-user NOMBRE_USUARIO	Busca los archivos de un determinado usuario.
-size +-tamaño[unidad]	Busca los archivos mayores o menores de un determinado tamaño. Por defecto este tamaño se expresa en bytes, no obstante puede tomar los siguientes valores: k -> Kbyte; M -> Mbyte; G -> Gbyte. Ej.- -size-20M --> Buscaría los archivos de menos de 20 Mbytes.
-group NOMBRE_GRUPO	busca los archivos del grupo indicado. Ej.- -group root
-d	muestra solo directorios y subdirectorios.
-exec comando{ } +	-exec comando{ } + ---> ejecuta un comando con cada fichero que encuentra. Ej.- find -exec rm{ } + (borraria todos los ficheros encontrados)

Comando file

Muestra el tipo de archivo pasado como parámetro.

Formato: file [OPCIONES] FICHERO

Comando grep

Busca un texto patrón en cada línea de un fichero de entrada.

Formato: grep [OPCIONES] patrón [FICHERO_ENTRADA]

Opciones básicas:

-w	Busca cadenas completas y no subcadenas, es decir la cadena patrón debe estar precedida y finalizada por blancos o fin de línea.
-c	Cuenta las líneas del fichero en las que se encontró la cadena patrón.
-i	No diferencia mayúsculas de minúsculas.
-r	Busca de forma recursiva
-n	Muestra la línea (y su nº) que contiene el patrón.

Comando head

Muestra las 10 primeras líneas de un fichero.

Formato: head [OPCIONES] [FICHERO]

Opciones básicas:

-n NUMERO_LINEAS	Muestra el un número de líneas indicado.
------------------	--

Comando ln

Permite crear enlaces entre archivos.

Formato: ln [OPCIONES] FICHERO NOMBRE_ENLACE

Opciones básicas:

-s	Se crea un enlace de tipo simbólico. Esto es, el fichero que se crea no es un fichero en si, sino un acceso directo al fichero original, con lo cual, ahorramos espacio en disco, ya que el acceso directo ocupará menos. Los permisos de los ficheros que son enlaces simbólicos comienzan por la letra 'l'. Estos enlaces se denominan enlaces soft.
----	--

Notas:

En caso de no especificar la opción -s, lo que se obtiene es un enlace hard, es decir un fichero con el mismo tamaño que el original, pero que apunta a los mismos inodos, con lo cual, si modificamos uno, modificamos también el otro.

Para ver si un fichero tiene enlaces creados, basta hacer un ls -l sobre el fichero. El primer número que aparece es el número de enlaces sobre el fichero. Si tiene 1, no tendrá ningún enlace soft ni hard, ya que el 1 hace referencia al propio fichero.

Comando ls

Lista los ficheros y directorios de un directorio concreto.

Formato: ls [OPCIONES] [RUTA o FICHERO]

Opciones básicas:

-R	Lista los ficheros entrando en los subdirectorios recursivamente.
-C	Lista los ficheros en columnas.
-l	Lista también las propiedades y atributos.
-a	Lista ficheros incluidos los ocultos de sistema.
-t	Lista los archivos ordenados por fecha de modificación

Comando more

Muestra por pantalla un fichero de forma paginada.

Formato: more [FICHERO]

Comando mv

Mueve o renombra un directorio o fichero. Hay que introducir el fichero o ficheros a mover y un directorio destino.

Formato: mv [fichero2 .. ficheroN] destino

Comando mkdir

Permite crear un directorio.

Formato: mkdir nombre_directorio.

Comando nl

Muestra el contenido de un fichero con las líneas numeradas.

Formato: nl [FICHERO]

Comando pwd

Muestra el directorio en el cual nos encontramos. Este directorio se llama directorio actual o directorio activo.

Comando rm

Elimina ficheros o directorios.

Formato: rm fichero1 [fichero2 .. ficheroN]

Opciones básicas:

-r	Borra directorios recursivamente. Si no especificamos esta opción, rm sólo borrará ficheros
----	---

Comando rmdir

Permite eliminar un directorio vacío. Si el directorio no estuviese vacío, informa de esto por la salida estándar y no lo borra.

Formato: rmdir nombre_directorio

Comando sort

Muestra el contenido de un fichero en orden alfabético.

Formato: sort [OPCIONES] [FICHERO]

Opciones básicas:

-f	Ignora las mayúsculas y minúsculas.
-r	De mayor a menor.
-n	En orden numérico.
-u	Elimina de su salida las líneas iguales consecutivas.
-t 'CARACTER_DELIMITADOR'	Especifica el delimitador para la ordenación por columnas
-k NUMERO_COLUMNA	Ordena según la columna indicada. Previamente debe usarse la opción -t para indicar el carácter que indica el salto de columna.

Ejemplos:

sort -t ' ' -k2 f1 → Ordena el fichero f1 por su segunda columna usando el delimitador ' '

sort -t '?' -k3 f1 → Ordena el fichero f1 por su tercera columna usando el delimitador '?'

Comando tail

Muestra las 10 ultimas lineas de un fichero.

Formato: tail [OPCIONES] [FICHERO]

Opciones básicas:

-n NUMERO_LINEAS	Muestra el un número de líneas indicado.
------------------	--

Comando touch

Si el archivo existe actualiza sus fechas de modificación y acceso, si no, crea un archivo vacío con 0 bytes.

Formato: touch [OPCIONES] FICHERO

Opciones básicas:

-c	Si el fichero no existe, no lo crea.
-a	Modifica solo la fecha de acceso.
-t FECHA	Usa la fecha indicada en vez de la actual.
-m	Modifica solo la fecha de modificación

Comando tr

Elimina o reemplaza caracteres de una cadena de entrada.

Formato: tr [OPCIONES] [CADENA1] [CADENA2]

Opciones básicas:

-d 'cadena1'	Elimina todas las ocurrencias de 'cadena1' en el texto de un fichero de entrada.
-s 'cadena1' 'cadena2'	Sustituye todas las ocurrencias de 'cadena1' por 'cadena2' en el texto de un fichero de entrada. Si hubieran varias 'cadenas1' seguidas, se sustituirían por una sola 'cadena2'.

Ejemplos:

ls -l | tr -s ' ' ' ' → Muestra por pantalla el resultado del comando ls -l , pero si hay varios espacios en blanco seguidos los sustituye por uno solo.

ls -l | tr -s "[a-z]" "[A-Z]" → Muestra por pantalla el resultado del comando ls -l , pero cambiando todas las letras por mayusculas.

Comando uniq

Elimina las líneas repetidas de un fichero (o entrada estándar) mostrando sólo una en su salida por pantalla (o un fichero)

Formato: uniq [OPCIONES] [FICHERO]

Opciones básicas:

-c	Comienza la línea con el número de veces que aparece.
-i	Ignora mayúsculas y minúsculas al comparar.
-d	Muestra sólo líneas repetidas.
-u	Muestra sólo líneas repetidas.

Comando wc

Cuenta caracteres, líneas o palabras de un fichero. sin parámetros cuenta las líneas.

Formato: wc [OPCIONES] [FICHERO]

Opciones básicas:

-c	Cuenta los caracteres
-l	Cuenta las líneas
-w	Cuenta las palabras

Comando whereis

Muestra la ubicación del archivo ejecutable, página de ayuda y fuente de un comando del sistema. Esta búsqueda se realiza dentro de los directorios indicados en PATH.

Formato: whereis [OPCIONES] comando

Opciones básicas:

-b	Busca solo el ejecutable.
-m	Muestra solo el archivo de ayuda.
-s	Muestra solo el archivo fuente.

2.3 Comandos de gestión de procesos.

En este apartado se muestran los comandos que podemos utilizar para mostrar información o gestionar los procesos que se ejecutan en el sistema.

Comando at

Permite ejecutar tareas en una fecha y hora determinadas.

Formato: at [-f FICHERO] FECHA

Notas:

- Con la opción -f podemos introducirle un fichero que contendrá el comando a ejecutar en la fecha y hora indicada.
- Si no ponemos un fichero, nos aparece un prompt para que introduzcamos los comandos a ejecutar en la fecha especificada.
- La FECHA ha de seguir el siguiente formato: hh:mm mmddyy

Cada usuario puede programar sus tareas propias, las cuales se almacenarán en un fichero de sistema y se ejecutarán con sus permisos cuando les toque, no obstante, para que un usuario pueda ejecutar at, debe estar incluido en el fichero /etc/at.allow (si existe dicho fichero) y no incluido en el fichero /etc/at.deny (si existe dicho fichero).

El proceso que se encarga de lanzar las tareas registradas con at es el demonio /usr/sbin/atd

Comando atq

Muestra las tareas programadas con el demonio at

Formato: atq

Funcionamiento:

- Los datos que muestra son:
 - Identificador de la tarea.
 - Fecha de la tarea.
 - Usuario que la lanzó.

Comando atrm

Elimina una tarea programada con el demonio at

Formato: atrm IDENTIFICADOR_DE_TAREA

Comando bg

Pasa a estado background un proceso lanzado en primer plano y suspendido con CTRL+Z.

Formato: bg PID

Comando crontab

Permite programar tareas periódicas.

Mediante este comando podemos programar tareas periódicas que serán ejecutadas posteriormente por el demonio del sistema cron.

Cada usuario puede programar sus tareas propias, las cuales se almacenarán en un fichero personal para dicho usuario y se ejecutarán con sus permisos cuando les toque, no obstante, para que un usuario pueda ejecutar crontab, debe estar incluido en el fichero /etc/crontab.allow (si existe dicho fichero) y no incluido en el fichero /etc/crontab.deny (si existe dicho fichero)

Formato: crontab [-u USUARIO] [OPCIONES] [FICHERO]

Opciones basicas:

-u USUARIO	Trabaja con el fichero crontab del que se indica
-l	Muestra el contenido del fichero crontab del usuario
-r	Elimina el fichero crontab del usuario

Fichero:

El fichero de ordenes debe tener el siguiente formato en cada una de sus líneas.

minuto hora dia mes dia_semana comando

El día de la semana se especifica mediante un número del 0 al 7 donde: 0 y 7 equivalen a domingo; 1 es Lunes; 2 martes; 3 miércoles; 4 jueves y 5 viernes.

Además el carácter * lo activa todo y dos enteros separados por – indican un rango de valores.

Ej.- 10 23 12 11 * ps -ef --> Se realizará un ps -ef el día 12 del mes 11 a las 23:10

 * 23 12 11 * ps -ef --> Se realizará un ps -ef el día 12 del mes 11 a las 23 durante cada minuto de esa hora.

 1-5 23 12 11 * ps -ef --> Se realizará un ps -ef el día 12 del mes 11 a las 23 durante los primeros 5 minutos de esa hora.

Por ultimo indicar que si cargamos un fichero crontab, se borran los comandos existentes anteriormente, con lo cual es conveniente incluir dichas ordenes antiguas en el nuevo fichero.

Ejemplo de uso:

crontab -u guest FICHERO --> Carga el fichero que le pasamos como fichero de ordenes periódicas para el usuario guest.

Comando free

Muestra información sobre la memoria del sistema, tanto la principal como la de swap.

Formato: free [OPCIONES]

Opciones básicas:

-b	Muestra la memoria en bytes.
-k	Muestra la memoria en KB
-m	Muestra la memoria en MB
-g	Muestra la memoria en GB

Notas: La salida por pantalla de este comando nos muestra la siguiente información:

```
root@DebTest:~# free -m
              total        used         free       shared    buffers       cached
Mem:           249          35          213           0           6          17
-/+ buffers/cache:          12          236
Swap:          382           0          382
root@DebTest:~#
```

Los valores que aparecen son los siguientes:

En la fila **Mem:**

1. **Total:** Memoria total del sistema.
2. **Used:** Memoria total ocupada por los procesos o por páginas cacheadas en RAM.
3. **Free:** Memoria no utilizada.
4. **Shared:** Memoria usada por varios procesos a la vez.
5. **Buffers:** Memoria buffers de disco en RAM, para acelerar los accesos al disco duro.
6. **Cached:** Memoria caché de programas en RAM. Aquí se cargan páginas de programas que son usados habitualmente, aunque ahora no estén en uso. Con eso se aumenta la velocidad

del sistema cuando se usen dichos programas.

Si algún proceso necesita más memoria y no queda memoria libre, el sistema libera memoria de buffer o caché.

En la fila **Buffers/cache**:

1. **Used: Memoria real usada por los procesos del sistema.** Mem usada = used - shared - buffers - cached
2. **Free: Memora real libre del sistema.** Mem libre = free + shared + buffers + cached

En la fila **Swap**:

1. **Total:** Tamaño de la partición de swap.
2. **Used:** Espacio ocupado de swap
3. **Free:** Espacio libre en la partición de swap.

Comando kill

Envía una señal a un proceso.

Si este tiene preparado su manejador se señales para capturar dicha señal, ejecutará una rutina de interrupción que le haya asociado, en caso contrario finalizará su ejecución. Existen no obstante, señales que nunca pueden ser capturadas por el manejador, con lo cual forzarán la terminación del proceso.

Si no se especifica ningún nombre o código de señal, se envía la señal TERM, que finaliza el proceso.

Formato: kill [OPCIONES] PID

Opciones básicas:

-l	Muestra la lista de señales existentes.
- NOMBRE_SEÑAL	Envía la señal que se corresponde con el nombre que se indica.
-l NÚMERO_DE_SEÑAL	Envía la señal que se corresponde con el número que se indica.

Señales más usuales:

SEÑAL	CÓDIGO	ACCIÓN POR DEFECTO	NOTAS
SIGALRM	14	Mata al proceso	
SIGHUP	1	Mata al proceso	

SIGINT	2	Mata al proceso	
SIGKILL	9	Mata al proceso	Esta señal no puede ser capturada por el proceso
SIGPIPE	13	Mata al proceso	
SIGPOLL		Mata al proceso	
SIGPROF		Mata al proceso	
SIGTERM	15	Mata al proceso	
SIGUSR1		Mata al proceso	
SIGUSR2		Mata al proceso	
SIGVTALRM		Mata al proceso	
SIGSTKFLT		Mata al proceso	Esta señal no puede ser capturada por el proceso
SIGWINCH		Se ignora	
SIGCHLD		Se ignora	
SIGURG		Se ignora	
SIGSTOP			Esta señal no puede ser capturada por el proceso

Comando lsof

Muestra información sobre los ficheros abiertos por procesos.

Formato: lsof [OPCIONES]

Opciones básicas:

-c COMANDO	Lista los ficheros abiertos por un proceso
+D DIRECTORIO	Lista los procesos que están usando mi directorio

-i:PUERTO	Lista los procesos que hacen uso de un puerto.
-----------	--

Comando ps

Muestra los procesos que se ejecutan en el sistema. Si no se introducen parámetros, muestra los procesos lanzados por cualquier usuario en la shell actual.

Formato: ps [OPCIONES]

Opciones básicas:

-u NOMBRE_USUARIO	Muestra los procesos de un determinado usuario
-e	Muestra todos los procesos existentes en memoria, sean del usuario o shell que sean. Si no se indica esta opción solo se mostraran los procesos lanzados en el terminal en el que nos encontremos.
-f	Muestra información más detallada de los procesos. Concretamente las siguientes columnas: <ul style="list-style-type: none">• UID → Identificador del usuario propietario del proceso.• PID → Identificador del proceso.• PPID → Identificador del proceso padre.• C → Porcentaje de uso actual de la CPU• STIME → Hora de lanzamiento del proceso.• TTY → Terminal desde el que se lanzó el proceso. aquí podemos tener dos valores:<ul style="list-style-type: none">• ttyNUMERO → Indica que el proceso se lanzo en un terminal de modo comando.• pts/NUMERO → Indica que el proceso se lanzo desde una consola del modo gráfico.• TIME → Tiempo de uso de CPU del proceso• CMD → Comando que se está ejecutando
u	Muestra ciertos datos interesantes de los procesos, como por ejemplo el estado de este en la columna STAT. Esta columna puede tomar uno o varios de los siguientes valores: <ul style="list-style-type: none">• Preparado (R).- Proceso que está listo para ejecutarse. Simplemente está esperando a que el sistema operativo le asigne un tiempo de CPU.• Ejecutando (O).- Sólo uno de los procesos preparados se está ejecutando en cada momento (monoprocesador).• Suspendido (S).- Un proceso se encuentra suspendido si no

	<p>entra en el reparto de CPU, ya que se encuentra esperando algún tipo de evento (por ejemplo, la recepción de una señal software o hardware). En cuanto dicho evento se produce, el proceso pasa a formar parte del conjunto de procesos preparados.</p> <ul style="list-style-type: none">• Parado (T).- Un proceso parado tampoco entra en el reparto de CPU, pero no porque se encuentre suspendido esperando algún evento. En este caso, sólo pasarán a estar preparados cuando reciban una señal determinada que les permita continuar.• Zombie (Z).- Todo proceso al finalizar avisa a su proceso padre, para que éste elimine su entrada de la tabla de procesos. En el caso de que el padre, por algún motivo, no reciba esta comunicación no lo elimina de la tabla de procesos. En este caso, el proceso hijo queda en estado zombie, no está consumiendo CPU, pero sí continua consumiendo recursos del sistema.• < .- El proceso tiene una prioridad alta.• N.- El proceso tiene una prioridad baja.• L.- El proceso tiene páginas bloqueadas en memoria.• s.- El proceso es el líder de la sesión. El proceso líder es el padre del resto de procesos de esa sesión, es decir el de PID más bajo. Normalmente este proceso sera el proceso shell (bash).• +.- El proceso fue lanzado en primer plano.• l.- El proceso es un thread o hilo de otro.
--	---

Comando pstree

Muestra un árbol con la jerarquía de los procesos existentes en memoria.

Formato: pstree [OPCIONES]

Opciones básicas:

-p	Muestra el PID de cada proceso.
----	---------------------------------

Comando sleep

Hace una pausa de tiempo en un determinado shell. Suele usarse para retener la ejecución de algún comando en shellscrip.

Formato: sleep NUMERO DE TIEMPO[SUFIJO]

Opciones básicas: El sufijo indica si el NUMERO se refiere a:

h	horas
m	minutos
s	segundos

Si no se especifica nada se toma como unidad el segundo.

Comando top

Muestra información sobre los procesos que se están ejecutando. Funciona como el administrador de tareas de Windows.

Formato: top

Comando watch

Ejecuta un comando de forma periódica mientras este abierto el shell

Formato: watch [OPCIONES] COMANDO

Opciones básicas:

-n SEGUNDOS	Ejecuta el comando cada 'vez que se cumplen los segundos indicados
-------------	--

2.4 Comandos de gestión de usuarios.

Seguidamente trataremos los principales comandos que necesitaremos para crear y gestionar usuarios y grupos.

Comando addgroup

Permite añadir un grupo al sistema. Para realizar esta operación hay que ser root.

Como resultado de este comando se crea una nueva entrada en el fichero `/etc/group` para registrar dicho grupo.

Formato: `addgroup [OPCIONES] NOMBRE_GRUPO`

Opciones básicas:

<code>--gid NUMERO_GID</code>	Fuerza a que se asigne a ese grupo un determinado número como identificador de grupo
<code>--system</code>	Crea un grupo de sistema.

Nota: Este comando realiza las acciones de creación, según las variables de entorno que se especifican en el fichero `/etc/adduser.conf`

Comando adduser

Permite añadir usuarios al sistema. Para realizar esta operación hay que ser root.

Como resultado se crea:

- Una entrada en el fichero `/etc/passwd` para registrar al usuario.
- Una entrada en el fichero `/etc/shadow` para almacenar la password del usuario.
- El directorio `$HOME` del usuario con sus ficheros ocultos de configuración.

Los nuevos usuarios del sistema tendrán como shell `/bin/false` (a menos que se modifique con la opción `--shell`), y tienen la clave deshabilitada.

Formato: `adduser [OPCIONES] NOMBRE_USUARIO`

Opciones básicas:

<code>--uid NUMERO_UID</code>	Fuerza a que se asigne a ese usuario un determinado número como identificador de usuario.
<code>--system</code>	Crea un usuario de sistema.
<code>--home DIRECTORIO</code>	Usa el directorio indicado, como directorio <code>\$HOME</code> del usuario, en vez del predeterminado en la variable <code>DHOME</code> del fichero <code>/etc/adduser.conf</code> .

--shell SHELL	Usa el shell que se indica para el usuario en vez del predeterminado en el fichero /etc/adduser.conf
--ingroup GRUPO	Añade el nuevo usuario al GRUPO en vez de al grupo del usuario o el grupo predeterminado definido por USERS_GID en el fichero /etc/adduser.conf.
--add_extra_groups	Añade al usuario a los grupos que se indican en la variable EXTRA_GROUPS del fichero /etc/adduser.conf.
--no-create-home	No crea el directorio \$HOME del usuario. Esta opción suele usarse en la creación de usuarios de sistema.
--disabled-login	No ejecuta passwd para establecer la clave. El usuario no podrá usar la cuenta hasta que se establezca una clave con el comando passwd. Además, en el fichero /etc/shadow aparecerá el carácter '!' en la posición de la password. A estos usuarios se les llama deshabilitados.

Nota: Este comando realiza las acciones de creación, según las variables de entorno que se especifican en el fichero /etc/adduser.conf

Comando chfn

Permite modificar los datos del usuario: nombre real, dirección, teléfono de casa y teléfono del trabajo.

Formato: chfn [OPCIONES] [USUARIO]

Opciones básicas:

-f NOMBRECOMPLETO	Modifica el nombre completo del usuario.
-r DIRECCION	Modifica la dirección del usuario.
-w TELEFONO_DE_TRABAJO	Modifica el número de teléfono del trabajo.
-h TELEFONO_DE_CASA	Modifica el número de teléfono de casa.

Nota: Por defecto el usuario sobre el que se opera es el que está ejecutando el comando en el shell. No obstante, root podría especificar otro usuario y modificar sus datos. Hay que indicar también que todos estos datos se almacenan en la línea de ese usuario, dentro del fichero /etc/passwd

Comando chsh

Permite modificar el shell del usuario.

Formato: chsh [OPCIONES]

Opciones básicas:

-s SHELL	Cambia el shell del usuario al que se pasa como parámetro. Ej.- -s /bin/bash
----------	--

Nota: Por defecto el usuario sobre el que se opera es el que está ejecutando el comando en el shell. No obstante, root podría especificar otro usuario y modificar sus datos.

Hay que indicar también que este dato se almacena en la línea de ese usuario, dentro del fichero /etc/passwd

Comando delgroup

Permite eliminar un grupo.

Formato: delgroup [OPCIONES] GRUPO

Opciones básicas:

--only-if-empty	Solo borra el grupo si no tiene ningún usuario dentro. Si no se especifica esta opción, se tendrá en cuenta el valor de la variable ONLY_IF_EMPTY del fichero /etc/deluser.conf
-----------------	---

Notas: Es necesario ser root para ejecutar este comando.

Además, no se puede borrar un grupo si es grupo primario (su GID está en el campo GID de la fila de ese usuario en el fichero /etc/passwd) de algún usuario.

Comando deluser

Permite eliminar un usuario.

Formato: deluser [OPCIONES] USUARIO

Opciones básicas:

-- remove-home	Elimina el directorio \$HOME del usuario. Si no se especifica se actuará según la variable REMOVE_HOME del fichero /etc/deluser.conf
--remove-all-files	Busca en el sistema todos los ficheros de ese usuario y los elimina. Si no se especifica se actuará según la variable REMOVE_ALL_FILES del fichero /etc/deluser.conf
--backup	Hace una copia de seguridad de los ficheros borrados con alguna de las opciones anteriores. Empaqueta todos los ficheros a borrar en un archivo llamado NOMBRE_USUARIO.tar. Si no se especifica se hará la copia de seguridad o no, según el valor de la variable BACKUP del fichero /etc/deluser.conf. Esta opción no funciona si no se pone también --remove-home o --remove-all-files

<code>--backup-to</code>	Deja la copia de seguridad de los ficheros a borrar, en el directorio indicado. Si no se especifica se usará el directorio indicado en la variable <code>BACKUP_TO</code> del fichero <code>/etc/deluser.conf</code> . Esta opción debe ir siempre acompañada de <code>--backup</code> para funcionar correctamente
--------------------------	--

Notas:

Es necesario ser root para ejecutar este comando.

Este comando realiza las acciones de borrado, según las variables de entorno que se especifican en el fichero `/etc/deluser.conf`

Comando groups

Muestra los grupos en los que está incluido un usuario

Formato: groups [USUARIO]

Si no se especifica el usuario, dará información sobre el usuario actual.

No es necesario ser root para ejecutar este comando.

Comando passwd

Permite modificar la password de un usuario.

Formato: passwd [OPCIONES] [USUARIO [CONTRASEÑA]]

Si no se especifica usuario ni contraseña, shell las pedirá por pantalla.

Comando su

Permite abrir un shell como otro usuario cuyo nombre se pasa por parámetro.

Formato: su USUARIO

Si ponemos sudo su estaremos intentando abrir una shell de root.

Hay que indicar que el sistema requerirá la password del usuario con el que queremos comenzar la nueva sesión.

Para cerrar la sesión escribiremos el comando **exit**

Comando umask

Establece los permisos por defecto que tendrán los nuevos ficheros creados por un usuario.

Formato: umask MASCARA_DE_PERMISOS_EN_OCTAL [USUARIO]

Notas:

Por defecto el usuario sobre el que se opera es el que está ejecutando el comando en el shell. No obstante, root podría especificar otro usuario y modificar su máscara de permisos.

La MASCARA_DE_PERMISOS funciona de la siguiente forma. Se calculan los permisos en octal que queremos que tengan los ficheros y luego se cambian los unos por ceros y los ceros por unos.

Hay que indicar también que umask sólo permite indicar los permisos de lectura y escritura, y no los de ejecución.

Ejemplo:

umask 777 ---> Quita todos los permisos.

umask 000 ---> Da permisos de lectura y escritura a todos.

umask 007 --> Da permisos de lectura y escritura al propietario y a los usuarios de grupo.

2.5 Comandos de empaquetado y compresión.

Comando add-apt-repository

Permite añadir un nuevo repositorio a las direcciones indicadas en el fichero /etc/apt/sources.list. Es necesario ser root para ejecutar este comando.

Formato: add-apt-repository ppa:nombre_de_repositorio

Ejemplo: add-apt-repository ppa:do-core/ppa

Comando apt-get

Permite instalar paquetes a través de Internet, para lo cual consultará en las direcciones indicadas en el fichero /etc/apt/sources.list. Es necesario ser root para ejecutar este comando.

Formato: apt-get [OPCIONES] [nombre_de paquete]

Opciones básicas:

install	Descarga un paquete de internet y lo instala. Ej.- apt-get install quota
remove	Desinstala un paquete pero no lo elimina del disco duro.
-f install	Repara dependencias errores de dependencias de paquetes
purge	Desinstala un paquete y lo borra del disco duro
autoremove	Se eliminan automáticamente los paquetes desfasados. Ej.- apt-get autoremove
update	Se usa para actualizar la lista de paquetes que tenemos instalados. Los índices de paquetes disponibles se obtienen de los lugares especificados en /etc/apt/sources.list. Siempre se debe realizar un update antes de un upgrade.
upgrade	Se usa para instalar la versión más nueva de todos los paquetes instalados en el sistema provenientes de alguna de las fuentes listadas en /etc/apt/sources.list. Los paquetes instalados con una nueva versión disponible son descargados y actualizados.

Ejemplos:

apt-get install quota

apt-get update

apt-get upgrade

Comando apt-cache

Permite buscar información sobre paquetes a instalar, a través de Internet, para lo cual consultará en las direcciones indicadas en el fichero /etc/apt/sources.list. Es necesario ser root para ejecutar este comando.

Formato: apt-cache [OPCIONES] [nombre_de paquete]

Opciones básicas:

search	Busca los paquetes asociados con un determinado concepto, usando para ello una cadena de búsqueda. Ej.- apt-cache search "atari"
depends	Muestra la información de dependencias de un determinado paquete
showpkg	Muestra una descripción de un paquete. Ej.- apt-cache showpkg NOMBRE_PAQUETE
pkgnames	Muestra toda la lista de paquetes que podemos instalar. Ej.- apt-cache pkgnames

Comando dpkg

Permite instalar paquetes con extensión .deb que previamente descargado al disco duro del pc. **No resuelve errores de dependencias.**

Formato: dpkg [OPCIONES] [nombre_de paquete]

Opciones básicas:

-i	Instala un paquete Ej.- dpkg -i avast.deb
-d	Desinstala un paquete Ej.- dpkg -d avast.deb
--get-selections	Lista los paquetes conocidos en el sistema y muestra su estado de instalación. Permite el uso de comodines para la búsqueda. Ej.- dpkg --get-selections dpkg --get-selections '*'
-p	Desinstala un paquete y elimina su contenido del disco duro Ej.- dpkg -p avast.deb
-l	Similar a la opción --get-selections, pero muestra información más completa. Concretamente tiene las siguientes columnas: Estado, nombre, versión,

arquitectura (32 o 64 bits) y descripción del paquete.

Significado de los caracteres de la primera columna:

El **primer carácter** de cada línea indica el estado deseado del paquete:

- u desconocido
- i instalar
- r eliminar
- p purgar
- h mantener

El **segundo carácter** de cada línea indica el estado actual del paquete:

- n no instalado
- i instalado
- c Sólo los archivos de configuración del paquete existen en el sistema.
- u El paquete sea desempaquetado, pero no se ha configurado
- f fallo configuración.
- h La instalación del paquete se ha iniciado pero no terminado por alguna razón.
- w El paquete esta a la espera del procesamiento de otro paquete.
- t El paquete depende del procesamiento de otros paquetes.

El **tercer carácter** de cada línea indica el estado de error del paquete:

- h mantener.
- r requiere reinstalar.
- x ambos problemas.

Comando gdebi

Permite instalar paquetes con extensión .deb que previamente descargado al disco duro del pc.

Intenta resolver errores de dependencias.

Formato: gdebi nombre_de paquete

Comando tar

Permite empaquetar y desempaquetar ficheros y directorios.

Manual básico de comandos Linux DEBIAN

Formato: tar [OPCIONES] ARCHIVO.tar [FICHERO1 FICHERO2...]

Opciones básicas:

-cvf	Empaqueta ficheros y directorios en un fichero de extensión .tar Ej.- tar -cvf pr1.tar D1 f1 f2 f3
-xvf	Desempaqueta el contenido de un fichero .tar Ej.- tar -xvf pr1.tar
-tvf	Muestra el contenido de un archivo .tar o .tgz Ej.- tar -tvf pr1.tar
-zcvf	Empaqueta archivos y directorios y los comprime obteniendo un fichero .tgz Ej.- Tar -zcvf pr1.tgz * En Linux, es posible generar un fichero .tar en vez del .tgz, no obstante es más correcto crearlo como .tgz para indicar que está comprimido.

Comando gpg

Permite encriptar y desencriptar un único archivo.

Formato: gpg FICHERO

Uso:

- Para encriptar usaremos la opción -c Ej.- gpg -c f1
- Para desencriptar no hace falta ninguna opción. Ej.- gpg f1.gpg

Notas: El archivo encriptado tendrá extensión gpg.

Comando gpg-zip

Permite encriptar y desencriptar varios ficheros o directorios.

Formato: gpg-zip [OPCIONES] [LISTA DE FICHEROS O DIRECTORIOS]

Opciones:

-c	Encriptación
-d	Desencriptación
-o	Especifica el fichero resultado de la encriptación.
--list-archive	Permite ver los nombres de los elementos encriptados en un fichero.

Ejemplos:

- `gpg-zip -c -o res.gpg *` → Esto encripta todos los ficheros y directorios de mi carpeta actual dentro del fichero `res.gpg`
- `gpg-zip -d res.gpg` → Desencriptación de los ficheros.
- `gpg-zip -l res.gpg` → Muestra los ficheros y directorios encriptados dentro de `res.gpg`

Comando gzip

Comprime y descomprime ficheros y directorios. Solo comprime ficheros si estos solamente tienen un link, es decir, si no se ha creado sobre ellos ningún enlace soft o hard.

Formato: `gzip [OPCIONES] FICHERO.tar`

Opciones básicas:

-d	Descomprime
-f	Fuerza la compresión aunque el fichero tenga algún enlace soft o hard.

2.6 Comandos de gestión de discos y dispositivos.

Comando df

Muestra el tamaño y espacio libre de los sistemas de ficheros del disco. Por defecto los muestra todos.

Formato: df

Comando dmesg

Muestra información sobre el log de alertas del kernel del sistema.

Formato: dmesg

Comando du

Indica la cantidad de espacio usada por un archivo o directorio

Formato: du FICHERO

Opciones básicas:

-h	Muestra la unidad del tamaño del fichero. KB, MB, GB, etc.
----	--

Comando fdisk

Permite crear, ver y modificar particiones en discos.

Formato: fdisk [OPCIONES]

Opciones básicas:

-l	Muestra las tablas de particiones de todos los discos del sistema.
-l DISPOSITIVO DE DISCO	Muestra las tablas de particiones de todos los discos del sistema.
DISPOSITIVO DE DISCO	Permite modificar las particiones de dicho disco. En este caso aparece por pantalla un menú con diferentes opciones.

Ejemplos:

`fdisk -l /dev/sda` ---> muestra las particiones del disco `/dev/sda`

`fdisk /dev/sda` ---> permite modificar las particiones del disco `/dev/sda`

`fdisk /dev/sdb` ---> permite modificar las particiones del disco `/dev/sdb`

Notas:

Es necesario ser root para ejecutar este comando, en caso contrario no veremos ninguna información. Además este comando muestra tanto particiones de Linux como de Windows o de otros sistemas.

Hay que indicar también, que Linux requiere de al menos dos particiones para funcionar. Estas son, una para el sistema de archivos y otra para swap.

El sistema de nombrado de dispositivos es el siguiente:

`/dev/sd[a-p]` --> para nombrar a los discos SCSI

`/dev/sd[a-p][1-4]` --> para nombrar particiones físicas en discos SCSI.

`/dev/sd[a-p][5-n]` --> para nombrar particiones lógicas en discos SCSI.

`/dev/hd[a-h]` --> para nombrar a los discos IDE

`/dev/hd[a-h][1-4]` --> para nombrar particiones físicas en discos IDE.

`/dev/hd[a-h][5-n]` --> para nombrar particiones lógicas en discos IDE.

Hay que indicar que un disco puede tener hasta 4 particiones físicas, de estas una será primaria y el resto extendidas. Por otra parte, dentro de estas, podemos crear tantas particiones lógicas como se requiera. Estas particiones lógicas comenzarán por el identificador 5 y de ahí en adelante. Seguidamente se añade un ejemplo para facilitar la comprensión del sistema de particiones:

Sea el siguiente disco:

| C: (1000Mb) | D: (500Mb) | E: (500Mb) |

En cuanto a particiones, este disco sería así:

Partición extendida

/ _____ \

| C: (1000Mb) | D: (500Mb) | E: (500Mb) |

\ _____ /

Partición primaria

Para Linux, el disco físicamente tendría la siguiente forma:

`/dev/hda1` -> 1000Mb -> Partición Primaria - C:

`/dev/hda2` -> 1000Mb -> Partición Extendida (D: + E:)

`/dev/hda5` -> 500Mb -> D:

`/dev/hda6` -> 500Mb -> E:

Es decir:

/dev/hda2

/ \

| /dev/hda1 | /dev/hda5 | /dev/hda6 |

Comando e2fsck

Permite revisar y reparar el sistema de ficheros de Linux (ext2, ext3 o ext4). Es necesario ser root para ejecutar este comando. Es importante que el sistema de archivos esté desmontado, si no, podemos crear graves daños.

Formato: fsck [OPCIONES] SISTEMA DE ARCHIVOS

Opciones básicas:

-p	Repara automáticamente los errores sin pedir confirmación
-n	Informa de los errores pero no los soluciona.
-y	Contesta “sí” a todas las peticiones de confirmación.
-v	Genera mas mensajes de diagnóstico

Comando lshw

Muestra información detallada sobre el hardware del equipo.

Formato: lshw

Comando lspci

Muestra información sobre todos los dispositivos conectados a las ranuras PCI del equipo.

Formato: lspci

Comando mkfs

Formatea una partición existente con un determinado sistema de ficheros. Si no se especifican

parámetros, se formatea el sistema de archivos con el tipo ext2. Es necesario ser root para ejecutar este comando.

Formato: mkfs [OPCIONES] SISTEMA_DE_FICHEROS

Opciones básicas:

-t TIPO_SISTEMA	Permite indicar el tipo de sistema de ficheros a utilizar. Por defecto será ext2. Entre los tipos de sistemas de formateo, tenemos: <ul style="list-style-type: none">• ext2• ext3• ext4• vfat• ntfs
-----------------	---

Ejemplos:

mkfs -t ext4 /dev/sdb1 ---> Formateo de la partición sdb1 en EXT4

mkfs -t vfat /dev/sdb1 → Formato de la partición sdb1 en FAT32

Comando mkswap

Permite crear una partición de tipo swap. Es necesario ser root para ejecutar este comando.

Formato: mkswap [OPCIONES] SISTEMA_DE_FICHEROS TAMAÑO_EN_KB

Opciones básicas:

-c	Comprueba la partición para verificar que no hay errores.
----	---

Ejemplos:

mkswap -c /dev/hda3 50000

Comando swapoff

Desactiva una partición swap. Es necesario ser root para ejecutar este comando.

Formato: swapoff PARTICION_SWAP

Comando swapon

Activa una partición swap. Es necesario ser root para ejecutar este comando. Este comando se ejecuta durante el arranque sobre todas las particiones de tipo swap en el fichero /etc/fstab

Formato: swapon PARTICION_SWAP

Comando mount

Permite montar dispositivos dentro de un directorio vacío del sistema de ficheros. De esta forma podemos entre otras cosas, poder añadir más discos al sistema, teniendo un solo árbol de directorios.

Si no ponemos ninguna opción, se muestran todos los sistemas de ficheros montados.

Formato: mount [OPCIONES] [FICHERO_DE_DISPOSITIVO] [PUNTO DE MONTAJE]

Opciones básicas:

-t	<p>Permite especificar el tipo de sistema de ficheros que se va a montar. Si no especificamos este parámetro, Linux intentará detectar el tipo de sistema de ficheros que debe montar. Los sistemas soportados por Linux, dependen del kernel. Los más comunes son:</p> <ul style="list-style-type: none">● auto (se detecta automáticamente el sistema de archivos que usa el dispositivo. Ante la duda es mejor usar esta opción).● ext2 (sistema nativo de Linux)● ext3 (mejora del estándar ext2)● iso9660 (formato de CD y DVD)● msdos (formato msdos)● ntfs (Formato NTFS, pero sólo permite lectura)● udf (para CD y DVD)● vfat (formato FAT de msdos)● tmpfs (temporal)● proc (específico para el directorio /proc)● devpts (para montaje de determinados dispositivos)● raisefs (sistema nativo de Linux)
-o	<p>Especifica una serie de opciones de montaje separadas por comas. Las principales son:</p> <ul style="list-style-type: none">● atime --> actualiza la fecha de acceso a cada inodo cuando se accede a este. Esta opción está por defecto.

- auto --> Este sistema podrá ser montado automáticamente con el comando mount -a
- defaults --> Se añaden las siguientes opciones por defecto: rw, suid, dev, exec, auto, nouser y async
- dev --> Indica que el sistema de ficheros puede almacenar ficheros de carácter o bloque.
- exec --> Permite ejecutar programas o ficheros en ese sistema de ficheros.
- noauto --> El sistema de ficheros no puede montarse automáticamente en el arranque del sistema. Solo podrá montarse de forma manual.
- nodev --> Indica que el sistema de ficheros no puede almacenar ficheros de carácter o bloque.
- noexec --> No se permite ejecutar programas o ficheros en ese sistema de ficheros.
- nouser --> Sólo root puede montar este sistema de archivos.
- owner --> Permite que un usuario que no sea root, pueda montarlo o desmontarlo, siempre que sea el propietario del fichero de dispositivo
- remount --> Si se desmonta el sistema de ficheros, se intentará montar otra vez automáticamente.
- ro --> Se monta el sistema de ficheros como de sólo lectura.
- rw --> Se monta el sistema de ficheros como de lectura y escritura.
- uid=USUARIO --> Establece al usuario indicado, como propietario de los ficheros del sistema de ficheros que vamos a montar. Si no hacemos esto, el propietario será root. Esta opción sólo es útil si montamos un sistema de archivos que no sea nativo de Linux, es decir, cualquier sistema menos raisefs, ext2, ext3 o ext4. En caso de montar un sistema nativo Linux, este parámetro se ignora, ya que los ficheros ya tienen sus usuarios propietarios asignados.
- user USUARIO --> Permite al usuario indicado montar y desmontar el sistema de ficheros.
- users --> Permite a cualquier usuario montar y desmontar el sistema de ficheros.

Ejemplos:

mount ---> muestra los sistemas de ficheros montados.

mount /dev/hda3 ---> busca en el fichero /etc/fstab, donde debe montar ese fichero de dispositivo y lo monta con las opciones que se indiquen.

mount -t ext3 /dev/hda4 /home2 ---> Monta la partición 4 del disco hda en el

directorio /home2, especificando que se trata de un sistema de tipo ext3.

`mount -t ext3 -o exec,users /dev/hda4 /home2 --->` Monta la partición 4 del disco hda en el directorio /home2, especificando que se trata de un sistema de tipo ext3.

Además realiza el montaje con las opciones `exec` y `users`.

Notas:

El fichero `/etc/fstab` almacena la información de los sistemas de ficheros que deben montarse en el arranque del sistema o al ejecutar el comando `mount -a`. El formato de este fichero es el siguiente:

`<dispositivo> <punto_de_montaje> <tipo_de_sistema_de_ficheros> <opciones> <dump> <pass>`

Cada uno de estos elementos se refiere a:

`<dispositivo> -->` Dispositivo a montar.

`<punto_de_montaje> -->` Ubicación donde se monta el dispositivo.

`<tipo de sistema de ficheros>`

`<opciones>`

`<dump> -->` Esta opción solo puede poseer el valor 0 o 1, en ella se guardan los errores en tiempo de sistema que ha reportado el sistema de archivos. Al activarlo (1) se hará un backup con las opciones, creando backup por si surge algún tipo de problema, lo normal es tenerlo desactivado (0), ya que rara vez se produce un error.

`<pass> -->` Esta opción puede poseer los valores 0, 1 o 2.

Con los valores (1 o 2) el sistema de archivos será revisado durante el proceso de arranque de forma automática con el comando `fsck`, para detectar y reparar errores lógicos. Los dispositivos que tengan un 1 se chequearan con más prioridad que los que tengan un 2.

En caso de estar desactivado (0), el sistema no realizara nunca ninguna pasada para comprobar la integridad del dispositivo físico.

Comando umount

Permite desmontar sistemas de ficheros.

Formato: `umount [DISPOSITIVO]`

Opciones básicas:

-a	Desmonta los sistemas de ficheros indicados en <code>/etc/mtab</code>
----	---

Ejemplos:

`umount /dev/hda2` ---> Desmonta la partición 2 del disco hda

`umount -a` ---> desmonta todos los sistemas de ficheros indicados en el fichero `/etc/mtab`

Notas:

El fichero `/etc/mtab`, mantiene los sistemas de ficheros montados en el sistema. Es mantenido automáticamente por el sistema, así que **NUNCA DEBEMOS MODIFICARLO MANUALMENTE**, ya que podrían producirse inconsistencias.

2.7 Comandos de gestión de red.

En el siguiente apartado, trataremos los comandos básicos que nos permitirán configurar y testear la red en un sistema Linux.

Comando ifconfig

Permite visualizar y modificar la configuración de red del sistema.

Formato: ifconfig [OPCIONES] [INTERFAZ] [CONFIGURACIÓN]

Opciones básicas:

-a	Muestra la información de las interfaces de red del sistema, estén activas o no.
----	--

Ejemplos:

ifconfig -a → Muestra los datos de los interfaces de red.

ifconfig eth0 netmask 255.255.255.0 192.168.0.20 → Configura la red de un interfaz

ifconfig eth0 down → Para el interfaz eth0

ifconfig eth0 up → Arranca el interfaz eth0

Notas: Para poner unos parámetros generales a las conexiones de red, debemos modificar el fichero **/etc/network/interfaces**. Por ejemplo podríamos tener:

```
auto lo eth0 eth1
iface lo inet loopback
iface eth0 inet static
address 192.161.1.60
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.1.1
iface eth1 inet dhcp
```

Aquí estaríamos indicando que:

- Tenemos tres interfaces de red lo, eth0 y eth1 (color azul)
- La interfaz lo es de tipo loopback es decir referencia al propio equipo, (color verde)
- La interfaz eth1 obtiene su configuración automáticamente por dhcp. (color amarillo)
- La interfaz eth0 configurada de manera estática (color rojo)

Comando ifdown

Desactiva una interfaz de red.

Formato: ifdown INTERFAZ

Ejemplos:

```
ifdown eth0
```

Comando ifup

Activa una interfaz de red.

Formato: ifup INTERFAZ

Ejemplos:

```
ifup eth0
```

Comando ping

Permite enviar paquetes de prueba a una máquina para comprobar si existe o no conectividad.

Formato: ping DIRECCIÓN_IP

Opciones básicas:

-I DEVICE	Permite especificar el interfaz de red por el que deben salir los paquetes.
------------------	---

Ejemplos:

```
ping 192.168.0.20
```

```
ping www.google.es
```

```
ping -I eth0 192.168.0.3
```