

Entrar en el sistema

Linux es un sistema multiusuario, por lo que para acceder a él se debe proporcionar un usuario válido como respuesta al comando **login**. Cada usuario posee una clave de acceso o *password* que impide la entrada a las personas ajenas al sistema. Una vez autenticado el sistema lanza un intérprete de comandos (*Shell bash*) para interactuar con el usuario.

Si la versión instalada es de tipo *escritorio*, se iniciará por defecto un entorno gráfico similar al empleado en Windows, sin embargo, existe la posibilidad de conectarse a un sistema *servidor* iniciado en un equipo a través de *emuladores de terminal*, como puede ser el programa *Putty* que es un cliente SSH/Telnet que permite la conexión al citado servidor iniciando una sesión en modo consola (texto).

- **who** muestra la lista de usuarios conectados.
- **whoami** nombre del usuario
- **hostname** nombre del equipo
- **uname -a** información del sistema
- **uname -r** versión del *kernel*
- **lsb_release -a** muestra la distribución instalada y su versión

Existe un usuario especial (*superusuario*) denominado *root*, encargado de la administración del sistema. Solo él puede realizar determinadas tareas:

- Configurar el sistema.
- Instalar un nuevo *kernel*.
- Añadir y eliminar usuarios y grupos.
- Montar sistemas de archivo para acceder al contenido de dispositivos externos (CD, USB, ...), aunque éste derecho puede proporcionarse a los usuarios incluyendo la opción *user* en la correspondiente entrada del fichero **/etc/fstab**.
- Desconectar el sistema.

Apagar el sistema

Para desconectar del sistema a un usuario basta con ejecutar la orden **logout**, sin embargo, el *root* debe desconectar de forma adecuada el sistema para no correr el riesgo de perder información o que el sistema de archivos quede dañado. Los comandos son:

- **shutdown -h now** o **halt** : para apagar el equipo.
- **shutdown -r now** o **reboot** : para reiniciar el sistema.
- **shutdown -h +10 "En 10 minutos se apagará el sistema" &** : envía un mensaje a todos los terminales y apaga el sistema en 10 minutos. El símbolo & al final de la instrucción indica que ésta se ejecutará en segundo plano.

Consolas virtuales

Hay disponibles 6 consolas virtuales desde las que se puede trabajar de forma independiente. Para pasar de una a otra se utilizan las combinaciones de teclas `<Ctrl+Alt>+<F1>...<F6>`, la *consola 2* (`<Ctrl+Alt>+<F2>`) está reservada para el entorno gráfico [X-Window](#), desde éste se puede acceder a las consolas de texto con la combinación `<Ctrl+Alt>+<F3>...<F6>`, en las que se puede trabajar de forma independiente. La *consola 1* (`Ctrl+Alt+F1`) es la de inicio.

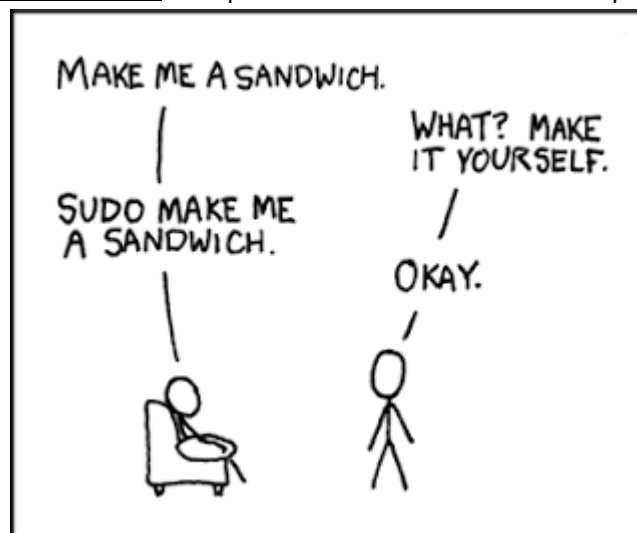
Si estás ejecutando Linux en un sistema gráfico, el atajo para ejecutar el Terminal suele ser `Ctrl+Alt+t`, con esta combinación mostrará en el escritorio el terminal desde el que podremos empezar a introducir comandos correspondientes al intérprete (*shell*) que se esté ejecutando; con el comando `echo $0` sabrás cual es.

```
alberto@alberto-Virtual-Machine: ~
Archivo Editar Ver Buscar Terminal Ayuda
alberto@alberto-Virtual-Machine:~$ uname -a
Linux alberto-Virtual-Machine 5.3.0-42-generic #34-18.04.1-Ubuntu SMP Fri Feb 28
13:42:26 UTC 2020 x86_64 x86_64 GNU/Linux
alberto@alberto-Virtual-Machine:~$ echo $0
bash
alberto@alberto-Virtual-Machine:~$
```

En este sistema el intérprete o Shell es **bash**.

El indicador al principio de la línea se denomina *prompt del sistema* y es configurable, aunque lo más habitual es que finalice con el símbolo `#` cuando estemos trabajando con el usuario *root* y sea un `$` si es un usuario normal.

Si un usuario pertenece al grupo *sudo* puede ejecutar comandos privilegiados, conociendo la contraseña del *root*, anteponiendo a cada comando la palabra **sudo**.



A pesar de que la aplicación “Actualización...” comprueba regularmente las actualizaciones, antes de instalar cualquier software conviene *actualizar los repositorios*¹ e *instalar las actualizaciones*² con los siguientes comandos de [apt](#):

sudo apt ¹update

sudo apt ²upgrade

Sistema de archivos

El sistema de archivos de Linux más habitual en la actualidad es el *ext4*, que es un sistema jerárquico cuya arborescencia está determinada por la *norma Filesystem Hierarchy Standard (FHS)*. Conocer esta estructura de directorios es esencial para la administración de un sistema Linux/Unix.

Directorios y nombres de archivo

El carácter para separar directorios en el camino de un archivo es */* al contrario que en Microsoft que es **.

El directorio raíz se representa con */*, y debe ser *montado* sobre una partición de un dispositivo concreto durante el proceso de instalación.

Ejemplo: */dev/sda1* → **sata disk a partition1**

UNIX distingue entre mayúsculas y minúsculas, lo cual se deberá tener en cuenta a la hora de introducir los comandos y los nombres de archivos y directorios.

Trabajar con directorios

Una vez dentro del sistema, el usuario se encontrará en su directorio *home*, cuyo nombre es posible conocer con el comando **pwd**, que muestra el nombre del directorio activo en el momento.

Para cambiar a otro directorio se emplea el comando **cd** *directorio*.

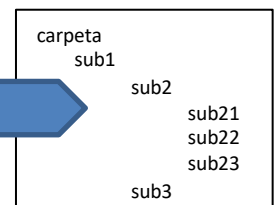


Respetar el espacio después del comando (**cd** *carpeta*). En un entorno de texto, a las carpetas se les suele denominar directorios.

- **cd /** ir al directorio *raíz*.
- **cd ..** subir al directorio *padre*.
- **cd ~** ir al directorio *home* del usuario.
- **cd** ir al directorio *home* del usuario.
- **cd -** ir al último directorio activo.

El carácter *~* puede sustituir al directorio *home* del usuario en una ruta. Las rutas pueden ser *absolutas* si comienzan a partir del directorio *raíz /*, de lo contrario son *relativas* al directorio activo. Se pueden emplear *comodines (Wildcards)* para hacer referencia a un conjunto de elementos que cumplan el patrón indicado.

- **mkdir [opción] directorio...** crea nuevos directorios.
 - **-p** crea la ruta completa.
- **mkdir -p carpeta/sub1/sub2/{sub21,sub22,sub23}/.././sub3**
- **rmdir [opción] directorio...** elimina directorios vacíos.
 - **-p** elimina la ruta completa.
- **rmdir -p uno/dos/tres** (solo si la ruta está vacía)
- **mkdir -p carpeta/{uno/{11,12},dos/{21,22,23}}**



➔ Instala el comando **tree**: `sudo apt install tree` para visualizar el árbol de un directorio.

➔ Puedes consultar la ayuda de un comando con [man comando](#) o buscarlo en [man bash](#)

Operaciones con archivos y directorios

- **touch *archivo*** crea un archivo vacío (para hacer pruebas)
- **echo "contenido del archivo" > archivo.txt** crea *archivo.txt*
- **ls [opciones] directorio/archivo...** muestra una lista de archivos y directorios
 - **-l** formato largo incluyendo permisos, propietario, fecha, etc.
 - **-t** ordenado por fecha y hora.
 - **-a** muestra incluso los archivos ocultos (su nombre comienza con *.*)
 - **-R** contenidos de subdirectorios
 - **-d** información de la entrada del directorio, no de su contenido.
- **cat *archivo*** muestra el contenido de un archivo.
- **more** y **less** muestran el contenido de un archivo con pausas entre páginas.
- **cp [opciones] origen... destino** copia *origen* en *destino*.
 - **-i** pregunta antes se sustituir *destino*.
 - **-R** incluye estructura y contenido de subdirectorios
 - **-f** fuerza la sustitución de *destino*.
 - **-b** realiza copia de seguridad de los sustituidos (*destino~*).
 - **-u** actualiza los archivos en el destino.
- **mv [opción] origen destino** mueve/renombra *origen* a *destino*.
- **mv [opción] origen... directorio** mueve archivos de la lista *origen* a *directorio*.
- **rm [opciones] directorio/archivo...** borra archivos o directorios.
 - **-R** elimina rutas completas, incluso no estando vacías.
 - **-i** pregunta antes de borrar.
- **find localización... [opciones] patrón** Busqueda
- **find . -name "a*"** lista los archivos cuyo nombre comience por *a* que estén en el directorio actual o cualquiera de sus subdirectorios.
- **find / -user fulano -group admin** lista todos los archivos cuyo propietario sea *fulano* y su grupo *admin*.
- **find . -type d -exec ls -ld {} \;** busca y lista todos los subdirectorios del directorio activo (inclusive). La siguiente sintaxis usando como filtro *xargs* sería más eficiente pues ejecutaría el comando *ls* sobre el conjunto del resultado de *find* y no con cada archivo de forma individual: **find -type d | xargs ls -ld**
- **du** muestra un sumario del uso de directorios.
 - **-h** muestra la información en formato "humano" (GB, MB, KB).
 - **-a** muestra información incluso de los archivos.
- **df** muestra información del sistema de archivos y sus puntos de montaje.
 - **-T** tipo del sistema de ficheros.
 - **-h** muestra la información en formato "humano" (GB, MB, KB).
- **sudo fdisk -l** lista la tabla de particiones.
- **sudo blkid** muestra los UUID (identificador único universal) de los dispositivos de bloques (unidades de disco, ...), que son de utilidad para que el administrador del sistema pueda utilizarlos en el archivo */etc/fstab* con la finalidad de "montar de forma automática" algunos dispositivos.
- **lsblk** muestra información sobre los dispositivos de bloques

La estructura de directorios y el propio sistema de archivos se “monta” a partir del directorio raíz /, incluso el acceso a medios de almacenamiento externo (discos duros, DVD, dispositivos USB, ...) se realiza a través de un directorio. A la operación de posibilitar el acceso a un medio se le denomina montaje y se realiza con el comando **mount**, asociando el dispositivo –representado por un archivo especial dentro del directorio */dev-* con un directorio existente normalmente dentro de */mnt* o */media*. Algunos dispositivos se montan automáticamente durante el inicio del sistema o se les asigna un punto de montaje predeterminado incluyendo la línea adecuada en el archivo */etc/fstab*, incluso se puede especificar si un dispositivo puede ser montado por un usuario sin privilegios. Una vez se haya terminado de trabajar con el dispositivo, se debe liberar con el comando **umount**. Solo usuarios con privilegios pueden *montar/desmontar* dispositivos.

Sintaxis: **mount -t sistemaArchivos /dev/archivoDispositivo /directorioPuntoMontaje**

- **sudo mount | grep 'ext4'** lista los dispositivos montados con el sist. de archivos *ext4*
- **sudo mount -t vfat /dev/sdb1 /media/usb** monta el dispositivo USB en el directorio */media/usb* ← (en *Hyper-V*, solo con *sesión mejorada*)
- **ls /media/usb** muestra el contenido del dispositivo USB
- **sudo umount /media/usb** libera el dispositivo USB.

Prueba a “insertar” la *iso* de instalación de Ubuntu en la unidad virtual DVD, después puedes acceder a su contenido montando el dispositivo *sr0* en el directorio */cdrom*:

- **sudo mount -t iso9660 /dev/sr0/cdrom** (se podría configurar en */etc/fstab* para que se establecieran algunas opciones por defecto). **-t hfs sería compat. con Mac**
- **ls /cdrom** muestra el contenido de la *iso* montada en este directorio.
- **sudo umount /cdrom**

```
sudo mount -t cifs //ip-equipowindows/recursoCompartido /mnt -o username=USUARIOWINDOWS
sudo mount -t cifs -o username=usuario,domain=dominio //ip/recurso /mnt
```

Enlaces simbólicos y físicos

Es posible crear en un directorio un enlace con un archivo en otra localización para facilitar su acceso y mantenimiento de copias, pues cualquier modificación realizada en un enlace o su original se actualiza en el resto.

- **ln rutaFichero rutaEnlace** crea una enlace duro o físico, con el mismo i-node
- **find / -inum 34567** busca los archivos cuyo i-node sea 34567
- **ls -li** muestra el i-node de cada archivo del directorio activo
- **ln -s rutaDirectorio rutaEnlace** para crear un enlace simbólico
- **ln -s ~/carpeta/archivo archivo.lnk** crea un enlace simbólico *archivo.lnk* con *archivo*. Observa que he especificado una ruta absoluta.
- **unlink rutaEnlace** elimina el enlace

La diferencia fundamental entre un enlace simbólico y otro físico radica en que, en el primer caso, una vez eliminado o movido el origen, desaparece el contenido de sus enlaces, por el contrario, un enlace físico conservaría su contenido hasta que se eliminara el último enlace. Otra diferencia importante es que se pueden crear enlaces simbólicos con directorios y con archivos en otra partición o en otro equipo (p.e. [NFS](#)).

Archivos empaquetados y comprimidos

Es posible empaquetar varios archivos o los archivos de un directorio en uno solo empleando el comando **tar**, que también se suele utilizar para hacer copias de seguridad.

- **tar [opciones] archivo.tar archivo/directorio...**
 - **c** crea un archivo .tar empaquetando los archivos/directorio dados.
 - **x** extrae archivos empaquetados en un archivo .tar
 - **t** lista los archivos contenidos en un .tar
 - **C** cambia al directorio destino para copiar el resultado.
 - **f** indica que se hace referencia a un archivo .tar y no a un dispositivo.
 - **P** guarda/restaura la ruta absoluta.
 - **r** añade un archivo a un paquete .tar existente.
 - **z** comprime/descomprime con *gzip* el archivo una vez empaquetado.
 - **j** comprime/descomprime con *bzip2* el archivo una vez empaquetado, se consigue una ratio de compresión más alta.
- **tar -cf datos.tar datos1 datos2 datos3**
- **tar -cf carpeta.tar carpeta**
- **tar -xf carpeta.tar -C ../Descargas**
- **tar -Pcf proyecto.tar /home/usuario/proyecto**
- **tar -tf carpeta.tar**
- **tar -Pxf proyecto.tar**
- **tar -rf paquete.tar archivo.txt**
- **tar -czf misDatos.tgz carpetaMisdatos**

Si solo se desea comprimir un archivo emplearemos **gzip** o **bzip2**:

- **-d** descomprime el archivo (igual que los comandos **gunzip** o **bunzip2**).
- **-r** comprime todos los archivos de un directorio (solo con *gzip*).
- **-k** conserva los archivos originales (solo con *bzip2*).
- **-dc** muestra por pantalla el contenido del archivo comprimido.
- **gzip f1** comprime el archivo f1 obteniendo f1.gz (se elimina f1).
- **gzip -dc f1.gz > f1** descomprime f1.gz obteniendo f1 (se mantiene f1.gz)

Para comprobar la integridad de un archivo (.tar, .iso, ...) podemos generar un *checksum* con **md5sum** o **sha256sum** (este último más seguro):

```
sha256sum paq.tar >paq.tar.sha256 ←genera el checksum
sha256sum -c paq.tar.sha256 ←verifica la integridad de paq.tar comprobando paq.tar.sha256
```

Control de procesos

Linux es un sistema *multitarea* y *multiusuario*. Cada usuario puede controlar la ejecución, pausa o destrucción de los procesos o tareas que ha creado.

Cuando se ejecuta un proceso el sistema le asigna un PID único que se puede utilizar para controlarlo.

- **ps** lista los procesos activos en el sistema (las opciones **-aux** muestran información más completa).
- **ps -aux | head** muestra los primeros procesos ejecutados por el sistema
- **pstree** muestra el árbol de procesos
 - **-p** incluye el PID del proceso
 - **-au** incluye información de los procesos del sistema.
- **top** muestra los procesos que más CPU consumen (salir con *q*).

Los procesos cuyo nombre termina en *d* suelen ser **daemons** o programas no interactivos residentes en memoria que se ejecutan en segundo plano, siendo "equivalentes" a los servicios de Windows.

Se pueden ejecutar procesos en segundo plano terminando la orden con el carácter &

- **find / -user huanfu >archivosHuanFu 2>/dev/null &** crea un archivo denominado *archivosHuanFu* con los nombres de los ficheros del usuario *huanfu*; en caso de producirse algún mensaje error, se dirigirá al dispositivo *null*, por lo que se omitirá. Como ejecuta el comando en segundo plano, podemos seguir operando en el terminal normalmente, de lo contrario bloquearía el terminal hasta la finalización del comando.
- **<Ctrl-z>** detiene un proceso que se está ejecutando en primer plano.
- **<Ctrl-c>** cancela un proceso que se está ejecutando en primer plano.
- **jobs** muestra las tareas del usuario en su sesión (*Shell*).
- **fg %numTarea** devuelve al primer plano la tarea.
- **kill** envía una señal a un proceso.
- kill -l** lista los números y nombres de las señales disponibles.
- kill -15** solicita al sistema detener un proceso enviando una señal SIGTERM.
- kill -9 PID** destruye un proceso conociendo su PID.
 - **pgrep nombreProceso** muestra su ID (puede varios ID, tantos como procesos con ese nombre)

kill %numTarea destruye una tarea conociendo su número de orden. Cuando un proceso no responde directamente a la orden *kill*, se puede forzar su destrucción con la señal -9 o -KILL.

Investiga el *daemon* [cron](#), el comando [crontab](#) y el archivo [/etc/crontab](#).

- ➔ **crontab -e** (*min hora día mes díaSemana comando*)
- ➔ Es necesario que esté configurado **timedatectl** para sincronizar con un servidor NTP. En nuestro caso **timedatectl set-timezone Europe/Madrid**
- ➔ Comprueba que *cron* está incluido en */etc/init.d* y está funcionando (**pgrep -l "\bcron\b"**)

Otros comandos

- **clear** borra la pantalla (<Ctrl+L>).
- **date** muestra la fecha y hora actual.
- **date +%A** muestra el día de la semana.
- **date +%T** muestra la hora actual.
- **cal** muestra el calendario del mes actual (-y muestra todo el año).
- **file archivo** informa del tipo de archivo.
- **whereis archivoEjecutable** informa de la localización de un archivo binario.
- **head** muestra las primeras líneas de un archivo.
- **tail** muestra las últimas líneas de un archivo.
- **grep [opción] patrón [archivo...]** localiza las líneas en archivo que contengan el *patrón* especificado. Algunas opciones: -i -c -r -v -o
grep -o -i "cadena" archivo.txt | wc -l muestra cuántas veces aparece *cadena* en el archivo
 → Admite [expresiones regulares](#).
grep "^import" *.java muestra las líneas de .java que empiecen con la palabra *import*.
 Caracteres para expresiones regulares básicas: ? * . [abc] [a-z] ^ \$
 Prueba `egrep 'a{2}' <<<"abracadabra"`
- **wc** cuenta caracteres (-m), palabras (-w) o líneas (-l) en un archivo.
- **who | wc -l** o **users | wc -w** muestra el número de usuarios conectados al sistema. En este caso el comando **wc** se utiliza como filtro del comando **who**.
- **cut -d " " -f1 archivo** muestra la primera palabra de cada línea de *archivo*.
- **cat /etc/passwd | grep 'bash' | cut -d ":" -f1** muestra los nombres de los usuarios que emplean *bash* como intérprete de comandos ([Shell](#)).
- **sort -t ";" -k 2 archivo** muestra el contenido de *archivo* ordenado por la segunda columna usando el carácter ; como separador de columna.
- **cat /dev/urandom | tr -dc '[:alnum:]' | head -c 12 >contraseña.txt** genera 12 caracteres alfanuméricos aleatorios y los guarda en el archivo contraseña.txt
- **sed -i 's/,;/g' archivo.csv** reemplaza todas las , por ; en *archivo.csv*
 → Busca más información sobre los comandos **tr** y **sed**
- **gpg -c contraseña.txt** encripta el archivo en contraseña.txt.gpg ([clave simétrica](#))
- **gpg -ca contraseña.txt** encripta el archivo en un archivo ASCII contraseña.txt.asc
- **gpg -d contraseña.txt.gpg** muestra el contenido del archivo (conociendo la clave)
 → Investiga sobre [GPG](#) y la [criptografía asimétrica](#).

```
gpg --full-generate-key
gpg -o clavePublica.gpg --export micorreo@servidor
gpg --import clavePublicaDestino.gpg
gpg -r correodestino@servidor.com --encrypt mensaje.txt
gpg -o mensajeDesencriptado.txt --decrypt mensaje.txt.gpg
```
- **su usuario** inicia un nuevo *shell* con la identidad de *usuario*.
- **su -c "comando opciones ..."** ejecuta el comando con la identidad del usuario *root*. Igual que **sudo**.
- **sudo adduser cuentaUsuario** script interactivo en lenguaje [Perl](#) propio de *Debian* que permite crear una cuenta de usuario.

Gestión de usuarios

Todos los usuarios de un sistema Linux están registrados en el fichero **/etc/passwd** y los grupos en **/etc/group**. Durante el proceso de instalación se crean algunos usuarios y grupos entre los que están el usuario y grupo **root**.

slice : x : 1002 : 1002 : Usuario Slice,,, : /home/slice : /bin/bash					
					Shell
				Carpeta personal	Ruta de la carpeta personal.
			Información del usuario	Nombre, ubicación, teléfono del trabajo, de la oficina.	
		ID de grupo (GID)	ID del grupo principal del usuario. La información de los grupos está en /etc/groups.		
		ID de usuario (UID)	El 0 está reservado para root y 1-99 para cuentas predefinidas. 100-999 para cuentas administrativas del sistema.		
	Contraseña	Una x indica que la contraseña se encuentra encriptada en /etc/shadow. Debe tener entre 6 y 8 caracteres como mínimo.			
Nombre de usuario	Nombre que identifica al usuario en el sistema. Debe tener entre 1 y 32 caracteres.				

Contenido del archivo **/etc/passwd**

Cuando se crea un usuario se toman determinados valores por defecto del archivo **/etc/default/useradd**, éstos pueden ser modificados con **useradd -D**. Un usuario puede pertenecer a varios grupos pero solo tendrá un *grupo principal* que será al que pertenezcan los archivos y directorios creados por el usuario.

Para crear un usuario se emplea el comando **useradd**:

- **useradd juancito** creará un usuario llamado *juancito*, cuyo grupo principal será *juancito* (que también se creará), no tendrá contraseña por lo que el sistema bloqueará la cuenta, su directorio de trabajo e intérprete de comandos se asignarán por defecto; pero no se creará su directorio de trabajo, el propio administrador deberá hacerlo posteriormente, así como asignar una contraseña de acceso con el comando **passwd juancito** para desbloquearlo (con este mismo comando se puede cambiar la contraseña más adelante).

Las contraseñas se guardan encriptadas en el archivo **/etc/shadow**.

slice:\$1\$NLJJ6\$ow5g1I1NgYITqqQQy5D21:14234:0:99999:7: : :					
				Caducidad	Días a los que se deshabilita la cuenta contados desde el 1 de enero de 1970.
				Inactivo	Días a los que se deshabilita la cuenta después de que caduque la contraseña.
				Aviso	Días a los que el usuario será avisado de que debe cambiar la contraseña antes de que ésta caduque.
				Máximo	Días durante los que la contraseña es válida. Al terminar el usuario tiene que cambiar la contraseña.
			Mínimo	Días que deben pasar como mínimo para que el usuario pueda cambiar la contraseña.	
			Último cambio	Días que han pasado desde la última vez que la contraseña fue cambiada contados desde el 1 de enero de 1970.	
	Contraseña	Contraseña encriptada. La forman entre 13 y 24 caracteres (a-z, A-Z, 0-9, \, /). Si comienza por el carácter \$, indica que la contraseña se ha encriptado usando un algoritmo distinto de DES. Si comienza por \$1\$, el algoritmo de cifrado está basado en MD5.			
Nombre de usuario	Nombre que identifica al usuario en el sistema. Debe tener entre 1 y 32 caracteres.				

Contenido del archivo **/etc/shadow**

.....

Opciones del comando **useradd**:

- **-c nombreCompleto** crea un comentario para la cuenta.
 - **-b dirBase** establece como directorio base para el *home* del usuario */dirBase/usuario*, si no será */home/usuario*. Incompatible con **-d**.
 - **-d directorio** especifica el *directorio home* de usuario.
 - **-e 2009-05-15** asigna la fecha de expiración de la cuenta (15/5/09).
 - **-g grupoPrincipal** establece el grupo principal del usuario.
 - **-N** no crea un grupo principal con el nombre del usuario, sino que se tomará el valor especificado en */etc/default/useradd*, y en caso de no estar definido el grupo principal del usuario será *users*.
 - **-G grupo2,grupo3,...** establece la pertenencia del usuario a uno a varios *grupos* adicionales.
 - **-m** crea el *directorio home* de usuario si no existe.
 - **-s Shell** asigna el intérprete de comandos que se iniciará tras el *login*.
 - **-k directoriok** copia los archivos y directorios contenidos en *directoriok* al *directorio home* de usuario, si no copiará el contenido de */etc/skel*.
 - **useradd -D** muestra las opciones por defecto
 - Las opciones **-b -e -g -s** son válidas en combinación con **-D** para asignar valores por defecto. **useradd -D -g GrupoDAM -s /bin/bash**
- **chage -l usuario** muestra/cambia opciones de expiración de la cuenta de usuario
 - **groupadd grupo** para crear grupos
 - **pwck** y **grpck** verifican la consistencia de los archivos */etc/passwd* y */etc/group* respectivamente.
 - **groups usuario** muestra la lista de grupos a los que pertenece *usuario*.
 - **id** muestra información sobre el usuario y grupos a los que pertenece.
 - **usermod opciones** modifica los atributos de cuenta de un usuario. Tiene las mismas opciones que el comando *useradd*.
usermod -a -G grupo usuario añade al usuario un nuevo grupo de pertenencia. Si el grupo es *sudo* convertimos al usuario en "*sudoer*".
 - **userdel -r usuario** elimina un usuario así como todo el contenido de su *directorio home*.
 - **deluser usuario grupo** eliminaría *usuario* del *grupo*.

Cambiar el propietario de un archivo:

- **chown juancito Informe09035** cambia el propietario del archivo *Informe09035*.
- **chown propietario:grupo archivo** cambia usuario y grupo propietario del archivo
- **chown -R juancito Presupuestos** cambia el propietario del directorio *Presupuestos* y de todos los archivos contenidos en él.
- **chgrp grupo archivo** cambia el grupo propietario de *archivo*.

Permisos y seguridad de archivos

El [sistema de seguridad de archivos de Linux](#) se basa en tres permisos (r, w, x) que pueden ser distintos para el usuario propietario, el grupo al cual pertenece y el resto de usuarios.

r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1
Propietario (u)			Grupo (g)			Otros (o)		

Archivos	Directorios
r = lectura w = escritura x = ejecución - = sin permisos	r = listar contenido w = crear/eliminar archivos o directorios x = entrar en el directorio (<i>cd</i>) - = sin permisos

Para modificar los permisos se emplea el comando *chmod*:

chmod [opción] modo[, modo]... archivo...

chmod [opción] modoOctal archivo...

-R cambia permisos en una ruta completa, compatible con el permiso X.

Modo simbólico: [ugoa]([-+=[rwxXst]))

Modo octal/numérico: número en octal de tres dígitos representando en binario los permisos para cada entidad (propietario, grupo, otros). (p.e.: 764 = 111 110 100 = rwx rw- r--)

chmod u+rwx,g+rw-x,o-rw+x archivo asigna o retira permisos.

chmod g=rw,o=r asigna de forma explícita permisos al grupo y otros.

chmod a+rwx archivo asigna todos los permisos a todos los usuarios.

chmod 764 archivo asigna los permisos **rwXrw-r--**

Permisos especiales:

SUID (s/S) – Si se asigna a un archivo, durante su ejecución toma la identidad de su propietario. **chmod u+s archivo**

SGID (s/S) – Si se asigna a un archivo, durante su ejecución toma la identidad de su grupo. Si se asigna a un directorio, los archivos o directorios que se creen dentro de él, heredarán el grupo de éste. **chmod g+s directorio**

Sticky-bit (t) – Cuando se asigna a un directorio, los archivos contenidos en él no podrán ser borrados o renombrados salvo por su propietario o por *root*. **chmod +t directorio**

.....

Cuando se crea un archivo o directorio, este adquiere unos permisos predeterminados que se obtienen de restar la máscara de usuario del máximo de permisos posibles (777 para directorios y 666 para archivos) el valor de esta máscara está determinado en el archivo `/etc/profile` o `~/profile` y suele ser 022 o también 0002, aunque puede ser modificado con el comando **umask**.

Listas de control de acceso

Las listas de control de acceso (ACL) [POSIX](#) son derechos de acceso detallados para archivos y directorios. Una ACL consta de entradas que especifican permisos de acceso a un objeto asociado. Las ACL se pueden configurar por usuario, por grupo o a través de la máscara de derechos efectiva.

Estos permisos se aplican a un usuario individual o a un grupo y usan los mismos permisos *rwX* que se encuentra en los permisos regulares del comando `chmod`.

[Antes de comenzar a trabajar con ACL](#), el sistema de archivos debe montarse con las ACL activadas. Esto se puede hacer en `/etc/fstab` para que los cambios sean permanentes. El soporte necesario viene habilitado de modo predeterminado en *CentOS*, *Fedora™*, *Red Hat™ Enterprise Linux*, *openSUSE™* y *SUSE™ Linux Enterprise*, y en versiones de *Ubuntu 14.04* o posteriores.

```
setfacl {-m|-x} [d:][u,g,o]:[usuario,grupo]:[r,w,x,octal1] {directorio|archivo ...}
getfacl {directorio,archivo}
```

¹ notación octal (4+2+1)

Opciones de setfacl

- m, --modify=acl** modifica ACL de un archivo.
- x, --remove=acl** elimina el ACL de un archivo.
- b, --remove-all** elimina todos los ACL de un archivo o directorio.
- k, --remove-default** elimina los ACL predeterminados.
- d, --default** Todas las operaciones se aplican sólo en el ACL predeterminado.
- R, --recursive** Aplica los cambios de ACL de modo descendente en un directorio.
- restore=archivo.acl** Restaura un respaldo de permisos obtenidos con **getfacl -R . >archivo.acl**
- test** Modo de pruebas. en lugar de cambiar los ACL de los archivos o directorios, sólo se listan los resultados.

- `setfacl -m u:usuario1:r archivo.txt` *usuario1* podrá leer el archivo
- `setfacl -m u:usuario2:6 archivo.txt` *usuario2* podrá leer y escribir
- `setfacl -x g:grupo1 archivo.txt` eliminar permisos al *grupo1*
- `setfacl -m d:g:grupo2:rw directorio1` establece los permisos predeterminados para los archivos creados en *directorio1*.
- `setfacl -b carpeta` elimina todas las ACL establecidas en *carpeta*

➔ Puedes probar los permisos establecidos sin necesidad de iniciar sesión en cada usuario con: `su usuario -c "comando"` que ejecutará *comando* con la identidad de *usuario*

Comandos de red

Las distintas Shell de Linux poseen multitud de comandos que permiten informar sobre el estado de la red, configurar y administrar esta.

- ➔ `domainname` muestra el dominio al que pertenece el equipo.
- ➔ `hostname [nombreEquipo]` muestra/establece el nombre del equipo.
- ➔ El archivo `/etc/hosts` contiene la correspondencia entre ip y nombre de equipo sin necesidad de utilizar un DNS.
- ➔ `ping direcciónIP` comprueba si hay conexión con el equipo con direcciónIP. Puede que el destinatario tenga bloqueada la respuesta al protocolo ICMP y no responda. Para detener el envío de paquetes, pulsa Ctrl+C.
- ➔ `ping direcciónIP -c 7` envía solo siete paquetes de prueba.
- ➔ `tracpath ip/host` rastrea la ruta para llegar a una ip o host. Es un comando equivalente a *traceroute*.
- ➔ `arp` muestra/modifica las tablas de traducción de dirección IP a dirección MAC.
- ➔ `dig dominio` recopila información del DNS. Equivalente a *nslookup*.
- ➔ `ip address show` muestra información del protocolo IP de las interfaces de red.
- ➔ `ip link show` muestra información de los dispositivos de red.
- ➔ `ip link set eth0 up/down` activa/desactiva una interfaz de red.
- ➔ `ip link set eth0 promisc on/off` activa/desactiva el modo promiscuo de una interfaz, que permite capturar todo el tráfico que circula por su red.
- ➔ `ip neigh show` muestra la tabla ARP (dispositivos próximos en el mismo enlace).
En *Powershell* ➔ `Get-NetNeighbor -AddressFamily ipv4`
- ➔ `ip addr add 172.16.0.24/16 dev eth2` establece la dirección IP para la interfaz eth2.
- ➔ En *Debian*, el archivo `/etc/network/interfaces` contiene la configuración IP de la red. Para asignar de forma estática la configuración de la interfaz *eth0*:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.25
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1 8.8.8.8
```

Después de editar este archivo habrá que renovar la IP con `sudo ip addr flush eth0` y reiniciar el servicio de red `sudo systemctl restart networking.service`

Para que tome la configuración del servidor DHCP: `iface eth0 inet dhcp`

En *Ubuntu*, a partir de 18.04 se utiliza `/etc/netplan/*.yaml`

- ➔ `dhclient -r` elimina la configuración establecida por el servidor DHCP.
- ➔ `dhclient` renueva la dirección IP tomándola del servidor DHCP.
- ➔ `tcpdump -i eth0 -A -w trafico.log` captura el tráfico de la interfaz eth0 y lo guarda en formato ASCII en el archivo trafico.log
- ➔ `tcpdump -r trafico.log` muestra el contenido del archivo obtenido.
- ➔ `ngrep -q "." "icmp"` muestra en pantalla el tráfico de red en el que aparezca la cadena icmp (protocolo asociado al comando *ping*).
- ➔ `wget url` permite descargar archivos de un sitio Web/FTP.
- ➔ `curl` alternativa a *wget* con más opciones. <https://curl.haxx.se/>