

Sistemas Informáticos

Unidad 05. Linux - Parte 2



Consejería d'Educació,
Investigació, Cultura i Esport

I.E.S. SERRA PERENXISA



UNIÓ EUROPEA

Fons Social Europeu

L'FSE inverteix en el teu futur

Autores: Sergi García, Alfredo Oltra

Actualizado Septiembre 2025



Licencia



Reconocimiento - No comercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

Importante

Atención

Interesante

ÍNDICE

1. Usuarios en Linux	3
1.1 Ficheros “/etc/passwd” y “/etc/shadow”	3
1.2 El comando sudo y la lista sudoers	3
1.3 El comando su	4
1.4 Creación de usuarios en Linux	4
2. Grupos en Linux	4
2.1 Fichero “/etc/group”	5
2.2 Crear grupos en Linux	5
3. Ficheros y directorios en Linux	5
3.1 Tipos de ficheros	5
4. Permisos en Linux	6
4.1 Algoritmo de concesión de permisos	7
4.2 Usando el comando “chmod” para establecer permisos	7
4.3 Permisos especiales	7
4.4 ACL (Access Control Lists) en Linux	8
4.5 Comandos básicos de ACL	8
5. Principales comandos de Linux	9
6. Bibliografía	12

UNIDAD 05. LINUX - PARTE 2

1. USUARIOS EN LINUX

Linux es un **sistema operativo multiusuario**. Cada usuario tiene un nombre asociado, pero **internamente es identificado por un número** llamado **UID** (*User Identifier*).

- Si dos usuarios tienen nombres distintos, pero el mismo UID, en realidad son **el mismo usuario** a nivel interno. 🙌 Más información: https://en.wikipedia.org/wiki/User_identifier

Tipos de usuarios

En Linux existen básicamente **dos tipos de usuarios**:

1. Usuarios normales

- Tienen **UID mayor que 0**.
- Sus operaciones están limitadas.
- Solo pueden acceder y modificar recursos para los que tengan permisos.

2. Usuario root (administrador)

- Tiene **UID = 0**.
- Es el **administrador principal** del sistema.
- Puede realizar prácticamente cualquier tarea: cambiar configuraciones, instalar programas, instalar controladores, ejecutar servidores, leer o borrar cualquier archivo, etc.

⚠️ Atención: trabajar como root puede ser muy peligroso. Un error puede dañar el sistema de manera irreversible. Si inicias sesión como root, debes saber exactamente lo que haces.

1.1 Ficheros “/etc/passwd” y “/etc/shadow”

En Linux, la información de los usuarios se gestiona mediante dos archivos principales:

- **/etc/passwd**: Contiene la lista de usuarios y atributos como el UID, el directorio personal (home), si el usuario está habilitado, etc.

Para ver su contenido se puede ejecutar:

```
cat /etc/passwd
```

Más información: <https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/>

- **/etc/shadow**: Contiene las contraseñas cifradas de los usuarios. Solo el usuario root puede leerlo o modificarlo.

Más información: <https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/>

1.2 El comando sudo y la lista sudoers

Aunque hemos visto que existen dos tipos de usuarios (root y normales), gestionarlos de forma estricta es ineficiente e inseguro.

Por eso, en distribuciones modernas como Ubuntu o Linux Mint, la cuenta root está desactivada por defecto (no se puede iniciar sesión como root directamente). Para este efecto, existe una lista llamada sudoers.

En esta lista se asignan privilegios especiales a usuarios normales. El privilegio más habitual es el de ejecutar temporalmente un comando como root mediante la orden sudo.

De esta manera, el sistema puede tener más de un administrador, sin necesidad de usar directamente la cuenta root.

Ejemplo: Si el usuario pepe (UID=1001) está en la lista de sudoers y ejecuta:

```
sudo cat fichero.txt
```

El comando cat fichero.txt se ejecutará con privilegios de root (UID=0).

💬 Curiosidad:

La primera vez que uses sudo en una sesión (o si pasó mucho tiempo desde el último uso), el sistema pedirá tu propia contraseña de usuario para confirmar tu identidad.

👉 Más información: <https://en.wikipedia.org/wiki/Sudo>

1.3 El comando su

El comando “su” significa Switch User (cambiar de usuario).

Se puede utilizar de dos formas:

- Sin parámetros: intenta iniciar sesión como root (UID=0).
- Con parámetros: seguido del nombre de usuario, permite cambiar a otro usuario.

Ejemplos:

- su pepe → el sistema intentará cambiar al usuario pepe.
- sudo su → el sistema intentará acceder como root.

👉 Más información: [https://en.wikipedia.org/wiki/Su_\(Unix\)](https://en.wikipedia.org/wiki/Su_(Unix))

1.4 Creación de usuarios en Linux

En Linux existen diferentes maneras de crear usuarios:

- Por línea de comandos (con herramientas como adduser o useradd).
 - <https://www.digitalocean.com/community/tutorials/how-to-add-and-delete-users-on-ubuntu-16-04>
- Con interfaz gráfica (por ejemplo, en distribuciones con herramientas de administración de usuarios).
 - <https://www.youtube.com/watch?v=DQHS1tQ2Xt8>

Cuando se crea un usuario, su directorio personal se genera automáticamente copiando el contenido de “/etc/skel”.

Este directorio funciona como una plantilla de configuración inicial para cada nuevo usuario.

👉 Más información: https://www.linfo.org/etc_skel.html

2. GRUPOS EN LINUX

Linux permite crear **grupos de usuarios**. Esto resulta muy útil cuando se quiere asignar permisos o privilegios a un conjunto completo de usuarios en lugar de hacerlo uno por uno.

Por ejemplo, se puede dar privilegios de “sudo” a un grupo. Así, cualquier miembro de ese grupo podrá usar el comando sudo para ejecutar operaciones como si fuera el usuario **root**.

💬 **Dato interesante:** en distribuciones como **Ubuntu** (y sus variantes como Lubuntu o Mint), pertenecer al grupo sudo significa automáticamente tener acceso a los privilegios de administración.

Un mismo usuario puede pertenecer a **varios grupos al mismo tiempo**, lo que permite una gestión flexible de permisos.

Al igual que los usuarios, los grupos tienen un **nombre** visible, pero internamente se identifican por un número entero llamado **GID** (Group ID). Si dos grupos comparten el mismo GID, en la práctica son el mismo grupo para el sistema.

2.1 Fichero “/etc/group”

La información de los grupos se guarda en el archivo **/etc/group**. Cada línea de este archivo representa un grupo e incluye:

- El nombre del grupo.
- El GID.
- La lista completa de usuarios que pertenecen a ese grupo.

👉 Más detalles sobre este archivo: <https://www.cyberciti.biz/faq/understanding-etcgroup-file/>

2.2 Crear grupos en Linux

- En consola, se puede usar el comando `groupadd` para crear un grupo y `usermod` para añadir usuarios.

```
sudo groupadd profesores
sudo usermod -aG profesores ana
```

- También se puede hacer de manera gráfica (según la distribución).
 - Tutorial en texto: <http://www.omniseccu.com/gnu-linux/redhat-certified-engineer-rhce/how-to-create-a-new-group-in-linux-using-groupadd-command.php>
 - Tutorial en vídeo: <https://www.youtube.com/watch?v=ZNeWntArcOg>

3. FICHEROS Y DIRECTORIOS EN LINUX

3.1 Tipos de ficheros

En Linux existen distintos tipos de archivos:

- **Archivos regulares:** contienen información (texto, imágenes, programas, etc.). Son los más comunes.
- **Directorios:** son archivos especiales que contienen referencias a otros archivos o directorios.
- **Enlaces:**
 - *Enlaces simbólicos (soft links):* apuntan a la ruta de otro archivo. Funcionan como los accesos directos de Windows. Si se borra el archivo original, el enlace queda roto.
 - *Enlaces duros (hard links):* no son un archivo diferente, sino un nombre adicional para un mismo archivo. El sistema los trata como idénticos, y el archivo no desaparece hasta que se borran **todas** las referencias.
- **Archivos especiales:** representan dispositivos físicos (discos, impresoras, etc.).
- **Archivos ocultos:** su nombre comienza con un punto (.), como `.bashrc`. No aparecen al listar directorios con `ls`, salvo que se use `ls -a`.

4. PERMISOS EN LINUX

Al ejecutar el comando:

```
ls -l
```

se obtiene una lista con los permisos de archivos y directorios.

Los permisos básicos son:

- **Lectura (r):**
 - En un archivo → permite leer su contenido.
 - En un directorio → permite ver qué contiene (listar con **ls**).
- **Escritura (w):**
 - En un archivo → permite modificarlo.
 - En un directorio → permite crear o borrar archivos dentro de él.
- **Ejecución (x):**
 - En un archivo → permite ejecutarlo como programa.
 - En un directorio → permite acceder a él con **cd**.

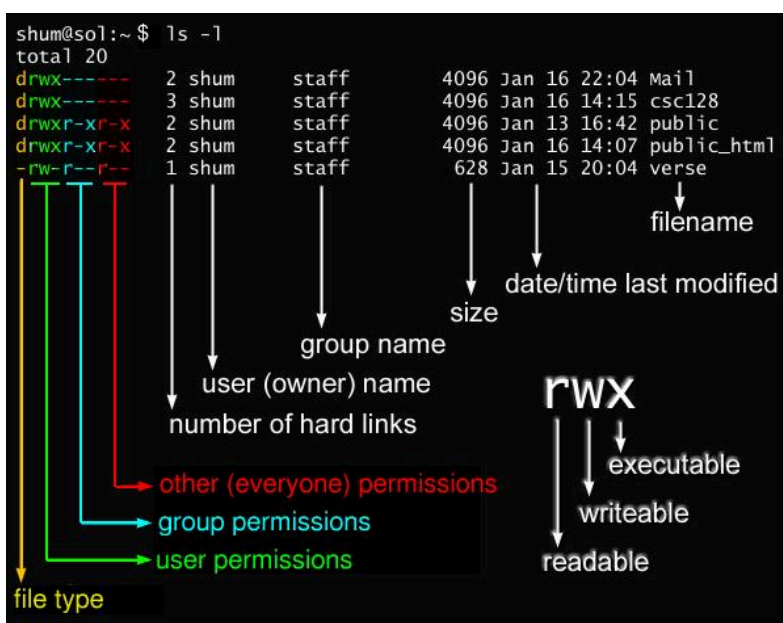
Estos permisos se definen para **tres categorías**:

1. **Propietario** (owner).
2. **Grupo** (group).
3. **Otros** (others).

Ejemplo de permisos con **ls -l** nos lleva a **-rwxr-x---**

- El propietario puede leer, escribir y ejecutar.
- El grupo puede leer y ejecutar.
- Los demás no tienen ningún permiso.

Un ejemplo de ejecución de “ls -l”



4.1 Algoritmo de concesión de permisos

Cuando un usuario intenta acceder a un recurso, el sistema sigue este orden de aplicación de permisos para determinar si se le concede el acceso o no:

1. Si el usuario es **root (UID=0)** → siempre tiene permiso.
2. Si el usuario es el **propietario**, se aplican los permisos de propietario.
3. Si no es propietario, pero pertenece al **grupo**, se aplican los permisos de grupo.
4. En cualquier otro caso, se aplican los permisos de **otros**.

👉 Es posible configurar permisos poco comunes, como que “otros” tengan más permisos que el propietario. Aunque raro, es totalmente válido.

4.2 Usando el comando “chmod” para establecer permisos

Para cambiar permisos se usa chmod. Solo el propietario del recurso o root pueden hacerlo.

Existen dos formas de usarlo:

- **Notación simbólica (alfabética):**

```
chmod u=rwx,g=rx,o= myFile.txt
```

→ El propietario tendrá todos los permisos, el grupo podrá leer y ejecutar, y los demás no tendrán ninguno.

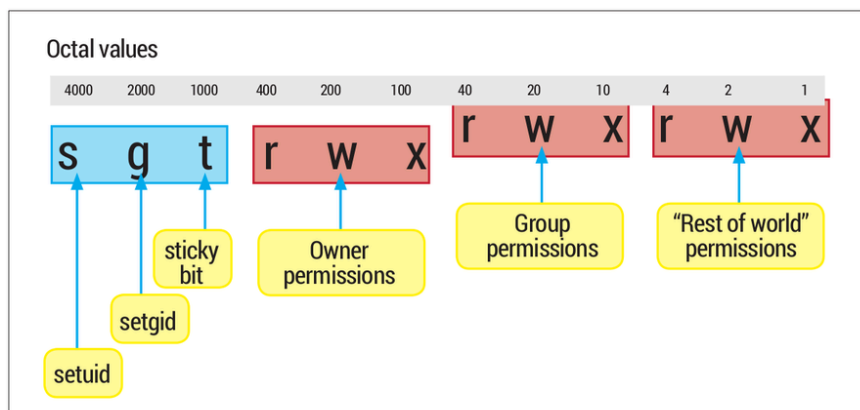
- **Notación octal:** Los permisos se representan en binario y luego en octal:
 - r=4, w=2, x=1.
 - Ejemplo: 750 = propietario 7 (rwx), grupo 5 (r-x), otros 0 (---).

```
chmod 750 myFile.txt
```

Más información sobre este comando en <https://itsfoss.com/es/comando-chmod-linux/>

4.3 Permisos especiales

Además de los 9 bits básicos, existen 3 permisos avanzados:



- **setUID (s en propietario):**
 - <https://en.wikipedia.org/wiki/Setuid>
 - En archivos ejecutables → el programa se ejecuta con los permisos del propietario, no del usuario que lo lanza.
 - En directorios → los nuevos archivos creados heredan el propietario del directorio.
- **setGID (s en grupo):**
 - <https://en.wikipedia.org/wiki/Setuid>
 - Igual que setUID, pero aplicado al grupo.
 - En directorios → los archivos nuevos heredan el grupo del directorio.
- **Sticky bit (t):**
 - https://en.wikipedia.org/wiki/Sticky_bit
 - Se usa en directorios compartidos (ejemplo: / tmp).
 - Permite que cualquiera pueda crear archivos, pero solo el propietario (o root) puede borrarlos, incluso si otros tienen permisos de escritura.

Más información sobre estos permisos en <https://srobb.net/setuid.html>

4.4 ACL (Access Control Lists) en Linux

En los sistemas modernos, además del modelo tradicional de permisos en Linux (propietario, grupo y otros), existen mecanismos más avanzados para gestionar la seguridad. Uno de los más importantes son las **ACL** (*Access Control Lists* o *Listas de Control de Acceso*).

Las ACL permiten definir **permisos más granulares** sobre archivos y directorios. Con ellas, no estamos limitados a los tres conjuntos clásicos (owner, group, others), sino que podemos asignar permisos específicos a **usuarios o grupos concretos** adicionales.

Por ejemplo:

- Un archivo puede pertenecer al usuario ana y al grupo profesores.
- Con el modelo clásico, solo podríamos definir permisos para:
 - ana (propietario).
 - profesores (grupo).
 - el resto de usuarios (otros).
- Con ACL, podemos dar permisos explícitos a un tercer usuario, por ejemplo maria, sin necesidad de cambiar el propietario ni el grupo del archivo.

4.5 Comandos básicos de ACL

Para trabajar con ACL en distribuciones modernas (Ubuntu, Debian, Fedora, etc.) existen utilidades como **getfacl** y **setfacl**:

- Ver las ACL de un archivo:

```
getfacl mi_archivo.txt
```

- Asignar permisos a un usuario concreto:

```
setfacl -m u:maria:rw mi_archivo.txt
```

👉 Esto permite que el usuario **maria** tenga permisos de lectura y escritura en el archivo, aunque no sea ni propietaria ni miembro del grupo asignado.

- Eliminar una ACL:

```
setfacl -x u:maria mi_archivo.txt
```

Las ventajas de las ACL son:

- Mayor flexibilidad que el sistema de permisos tradicional.
- Facilitan la gestión en sistemas multiusuario y entornos corporativos.
- Permiten herencia de permisos en directorios (útil para carpetas compartidas).

Consideraciones

- No todas las particiones están montadas con soporte ACL por defecto, aunque en distribuciones modernas suele estar habilitado.
- Pueden generar confusión si se mezclan con el modelo tradicional, ya que a veces los permisos mostrados con `ls -l` no reflejan las reglas adicionales de ACL.

Puedes profundizar en las ACL en Linux en estas guías:

👉 <https://www.tecmint.com/understanding-and-using-acls-in-linux/>

5. PRINCIPALES COMANDOS DE LINUX

En esta sección vamos a describir los comandos principales de la consola en sistemas Linux. Si quieres obtener información detallada sobre cualquiera de ellos, puedes usar el comando:

```
man comando
```

💬 **Interesante:** `man` muestra el manual/ayuda de otros comandos. Es muy útil y suele estar disponible en varios idiomas (inglés, español, etc.).

💬 **Interesante:** también resulta práctico tener una *Cheat Sheet* (chuleta de comandos). Existen muchas en Internet, por ejemplo:

👉 <https://linuxopsys.com/wp-content/uploads/2022/06/linux-cheat-sheet.pdf>

Comandos para gestionar la interfaz

Comando	Qué hace	Ejemplo
man	Muestra la ayuda de un comando.	man ls
clear	Limpia la pantalla de la terminal.	clear
echo	Muestra texto literal en pantalla.	echo "Hola Mundo"
exit	Cierra la sesión en la consola.	exit

Comandos para configurar el sistema

Comando	Qué hace	Ejemplo
---------	----------	---------

date	Muestra o cambia la fecha del sistema.	date → muestra la fecha sudo date -s "2025-09-18 14:00:00" → establece fecha/hora
cal	Muestra el calendario.	cal
shutdown	Apaga el sistema.	sudo shutdown now
reboot	Reinicia el sistema.	sudo reboot

⚠ Atención: para shutdown y reboot es necesario usar sudo en la mayoría de distribuciones modernas.

Comandos para obtener información sobre discos

Comando	Qué hace	Ejemplo
du	Muestra el uso de disco por archivo/directorio.	du -h (formato legible) du -sh * (resumen por carpeta)
df	Muestra información de los sistemas de archivos montados.	df -h

Comandos para gestionar archivos y directorios

Comando	Qué hace	Ejemplo
touch	Crea un archivo vacío o actualiza su fecha de modificación.	touch miArchivo.txt
nano / vi	Editores de texto en terminal.	nano miArchivo.txt vi miArchivo.txt
mkdir	Crea un directorio.	mkdir miDirectorio
cat	Muestra el contenido de un archivo de texto.	cat miArchivo.txt
more	Muestra el contenido de un archivo página a página.	more miArchivo.txt
less	Similar a more, pero más potente (permite desplazarse).	less miArchivo.txt
grep	Busca un patrón dentro de un archivo.	grep root /etc/passwd

ls	Lista el contenido de un directorio.	ls ls -la
cd	Cambia de directorio.	cd /home (ruta absoluta) cd ../miDir (ruta relativa)
pwd	Muestra la ruta actual.	pwd
rm	Borra archivos o directorios.	rm miArchivo rm -r miDirectorio
cp	Copia archivos o directorios.	cp miArchivo /home/admin/ cp -r miDir /home/admin/
mv	Mueve o renombra archivos/directorios.	mv antiguo.txt nuevo.txt
ln	Crea enlaces.	ln miArchivo hardLink.txt (enlace duro) ln -s miArchivo accesoDirecto.txt (enlace simbólico)
mount	Monta un dispositivo en una carpeta.	sudo mount /dev/sda1 /media/miDisco

⚠ Atención: rm -r es muy peligroso porque borra todo sin pasar por la papelera. Siempre revisa con ls antes de ejecutar un rm.

Comandos relacionados con permisos



Comando	Qué hace	Ejemplo
chmod	Cambia permisos de un archivo o directorio.	chmod 750 miArchivo
chown	Cambia propietario y/o grupo de un archivo o directorio.	sudo chown nuevoUsuario:nuevoGrupo miArchivo
chgrp	Cambia solo el grupo de un archivo.	sudo chgrp profesores miArchivo

Comandos de red

Comando	Qué hace	Ejemplo
ping	Comprueba la conexión con otra máquina.	ping google.com

ifconfig / ip a	Muestra información de red.	ip a
wget	Descarga archivos de Internet.	wget https://example.com/archivo.zip
curl	Transferencia de datos desde o hacia un servidor.	curl https://example.com

Para aprender más comandos puedes visitar los siguientes enlaces:

- Manual oficial de comandos GNU/Linux:
 -  <https://www.gnu.org/software/coreutils/manual/>
- Guía de referencia rápida de Linux (CheatSheet):
 -  <https://linux-training.be/files/books/html/linux-training.pdf>

6. BIBLIOGRAFÍA

[1] "The Linux command line" Libro Creative Commons <http://linuxcommand.org/tlcl.php>

[2] "Linux commands Handbook"

<https://bjpcjp.github.io/pdfs/devops/linux-commands-handbook.pdf>