

# Introducción a Linux y comandos en terminal.

---

Manual redactado por Hugo Ruiz Sánchez. Documento organizado por secciones, vista recomendada en procesadores de documentos que admitan esquemas.

## Introducción - Terminal, Bash y Shell

---

- El emulador de terminal permite el control del sistema operativo mediante una línea de comandos.
- Para este fin, el shell se encarga de interpretar los comandos recibidos y ejecutarlos.
- La shell más común en linux se llama Bash, y gestiona la terminal: por eso cuando sucede un error detalla "Bash: error ... "
- Cada vez que iniciamos el emulador de terminal, la Bash ejecuta un proceso.
- La inicialización de la Bash requiere de unas órdenes específicas, definidas en ~/.bashrc; estas permiten: crear alias de programas para automatizar tareas, personalizar el prompt para que informe de la hora en cada comando,...

## Variables de entorno

---

- Las variables de entorno condicionan el funcionamiento de los comandos y procesos (los procesos son instancias de comandos reproduciéndose)
- Una variable de entorno puede ser *PATH* \* *paraver* : \* \* \**echo*PATH\*\*, la localización donde se ubican los accesos de los programas para no tener que acceder directamente a ellos.
- Pueden consultarse las variables de entorno desde **env** y **echo (nombre\_variable)**
- Pueden crearse y editarse variables de entorno desde **export (nombre\_variable)=(valor)**
- Pueden borrarse las variables de entorno creadas con **unset (nombre\_variable)**

## Uso del manual

---

- **man (comando)** Facilita una guía de uso detallada para cada comando.

# Comando SUDO

---

El precedente "sudo" permite ejecutar comandos como administrador.

## Gestión de archivos.

---

### Espacio ocupado en el disco

- **df** Indica el espacio ocupado en el disco
- **df -h** human readable, permite una mejor visibilidad.
- **df -BM** ... en megabytes.
- **df -BG** ... en gigabytes.

### Espacio ocupado por las carpetas

- **du** indica el espacio ocupado por los directorios, es recursivo respecto a carpetas.
- **du -h** human readable, permite una mejor visibilidad
- **du -BM** ... en megabytes
- **du -BG** ... en gigabytes
- **du -d (*profundidad*)** limita la aparición de los archivos a una profundidad de carpetas, es decir, corta la recursividad para dar un informe concreto.

### Herramienta para ver espacio en el disco

- **ncdu** es una herramienta útil que se muestra en ventana de terminal, y permite ver con mayor facilidad el espacio ocupado por algunos archivos en el disco.

### Ubicación

- **pwd**: Indica la carpeta actual.
- **open .**: El gestor de archivos gráfico se abre en la misma carpeta.

### cd - Desplazamiento por carpetas.

- **cd (carpeta)**: Abrir una carpeta.
- **cd**: Devuelve a la carpeta de usuario.
- **cd ..**: Retorna a la anterior carpeta.

- **cd ../(directorio)** Acceder a un directorio ubicado en la anterior carpeta
- **cd ~(directorio)** El signo ~ simboliza la carpeta de usuario.

## ls - Listar archivos

- **ls**: Facilita una lista de archivos.
- **ls -a**: ... con ocultos.
- **ls -l**: Muestra más información (usuarios/grupo/tamaño/fecha)
- **ls -R**: Modo recursivo, lista el contenido de cada uno de los subdirectorios.
- **ls (carpeta)** Lista los archivos dentro de una carpeta.

## (apt install mlocate) locate - localizar archivos

- **sudo updatedb** actualizar la base de datos para llevar a cabo la búsqueda
- **locate ("\*.extension" / archivo)** Indica la localización de los archivos

## find - filtrar búsquedas

- **find (carpeta) -type d** directorios de la carpeta
- **find (carpeta) -type f** filtrar por archivos de la carpeta
- **find (carpeta) -type d** filtrar por directorios de la carpeta
- **find (carpeta) -name: (nombre)** filtrar por nombre del archivo (p.ej ".txt")
- **find (carpeta) -mmin (+/-minutos)** filtrar por minutos de creación (solo archivos con menos de x minutos de creación).
- **find ejemplo -mmin +5 -mmin -10** busca archivos en la carpeta *ejemplo* entre 5 y 10 minutos
- **find (carpeta) -mtime (+/-minutos)** filtrar por días de creación (solo archivos con menos de x días de creación).

**find (carpeta) -maxdepth 1** máxima profundidad de carpetas, pone limite a la recursividad

- **find (carpeta) -size (+/-tamaño(k/m/g))** filtrar por peso del archivo
- **find (carpeta) -empty** solo registra archivos vacíos

- **find (carpeta) -perm (permiso\_numerico)** filtrar por el permiso

## Borrado con find

- **\*find . -type f -name ".txt" -exec rm -rf {} + \*\*** Gracias a find pueden borrarse los archivos en base a su extensión, por ejemplo. Para esto será necesario un **-exec (comando)** {placeholder} + significando el exec el indicativo a partir del cual se ejecuta un comando, el placeholder la salida recibida antes del -exec y el + la sucesión de todas las salidas
- **!find** realiza la última búsqueda hecha con find

## Permisos

- Tanto archivos, usuarios y grupos pueden tener permisos de lectura (r), escritura (w) y ejecución (x).

## ls -l Ver permisos

- **ls -l** Permite ver información acerca de los permisos, separada en permisos, usuario y grupo.
- **ls -l (archivo)** ... para un archivo concreto.
- Partiendo del ejemplo -rw-rw-r--
  - El primer "-" significa que el archivo es un fichero normal. Si es una "d", se trata de una carpeta.
  - Las tres secuencias de caracteres sucesivas (rwx) corresponden respectivamente a usuarios, grupo, y otros - cualquiera - en un orden numérico de 421, significando su suma total (7) la totalidad de los permisos, necesario para modificarlos con el comando chmod.

## Asignar permisos a los archivos

- **chmod (numeropermiso) (archivo)** Permite la modificación de los permisos en base a la suma de los numeros.
- **chmod -x (archivo)** Quita permiso de ejecución de todas las filas
- **chmod g-x (archivo)** Quita permiso de ejecución solo al grupo
- **chmod a+rwx (archivo)** Concede todos los permisos al archivo (777); la "a" es de "all"

- **chmod u=(letras),g=(letras),o=(letras) (archivo)** Permite una fácil asignación de los permisos. Todo debe estar junto.

## Usuarios y grupos

- **getent** permite consultar las bases de datos que almacenan:
  - **getent passwd** Lista de todos los usuarios
  - **getent group** Lista de todos los grupos
  - **getent hosts** Lista de todos los hosts

## Grupos

- Los grupos de usuario son distintos tipos de personas que usaran el equipo, y para los cuales se adecúan permisos distintos.
- **whoami** indica a qué grupo pertenece el usuario. Si ejecutamos el comando como root, arrojará que somos "root".
- **chown (grupo) (archivo)** limita el acceso de un archivo a un grupo

## Usuarios

- **useradd (nombre\_de\_usuario)** Añade un nuevo usuario.
- **passwd (contraseña)** Añade una nueva contraseña a ese usuario
- **userdel (nombre\_de\_usuario)** Elimina el usuario
- **id** Permite ver usuario y grupos a los que pertenece

## Manejo de archivos de texto

### Lectura de ficheros

- El comando **cat** significa concatenar, es decir, por si solo únicamente permite escribir texto en terminal sin interrumpir el flujo.
- **cat (archivo) (archivo1) (archivo2) ...** Lectura directa en terminal de un archivo de texto
- **sed -n (numero)\_p \_(archivo)** Ver solo la línea indicada, por ejemplo sed -n 5p archivo.txt
- **sed -n -e (numero)\_p -e \_(numero)\_p ... \_(archivo)** Para indicar unas líneas determinadas
- **sed -n (numero),(numero)\_p \_(archivo)** Para indicar un rango, ejemplo: sed -n 5,8p archivo.txt ; pueden hacerse ambos anteriores conjuntamente, por ejemplo sed -n -e 5,8p -

e 10p archivo.txtzzzzz

- **head -n (numero) (archivo)** Enlista un límite de líneas escritas en un fichero desde el principio, cabeza.
- **tail -n (numero) (archivo)** Enlista un límite de líneas escritas en un fichero desde el final, cola.
- **cut -c (rango1-rango2) (archivo)** Corta los caracteres en el rango indicado para cada línea, por ejemplo cut -c 2-3 fichero.txt

#### Lectura ordenada de ficheros

- **diff (archivo) (archivo)** Compara dos archivos línea por línea para apreciar variaciones.
- **sort \_(fichero)** Muestra las palabras escritas ordenadas alfabéticamente.
- **sort -r \_(fichero)** ... en sentido inverso.
- **sort -n \_(fichero)** ... por número
- **sort -f \_(fichero)** ... sin importar mayúsculas y minúsculas

#### Filtrar ficheros por contenido

**-grep ("texto") (archivo.txt)** Indica la presencia de un determinado texto en un fichero, aunque también puede hacerse sobre todos los archivos (\*.extension), (\*) un (directorio), o sobre un (directorio/\*.extension). Puede recorrerse la búsqueda por todos los directorios usando -r y seleccionando el directorio a partir del cual hacer la recursión : **grep -r (texto)** .

**grep -e ("texto") -e ("texto") -e ("texto") (archivo.txt)** Permite hacer un grep con varios elementos a la vez.

- **grep -n ("texto") (archivo.txt)** ... indica además la línea donde se encuentra.
- **grep -i ("texto") (archivo.txt)** ... dejan de importar mayúsculas y minúsculas y busca la repetición de caracteres sea donde sea
- **grep -l ("texto") (archivo.txt)** ... indica el directorio donde se encuentra
- **grep -c ("texto") (archivo.txt)** ... cuenta la cantidad de coincidencias en cada archivo.
- **grep -B (numero) ("texto") (archivo.txt)** Inica un determinado número de líneas previas a la coincidencia

- **grep -p ("expresión regular") (archivo)** Grep tiene una función compatible con Perl Regular Expressions, lo que permite un mejor filtrado (<https://cheatography.com/davechild/cheat-sheets/regular-expressions/>)
- **grep -P ("expresión regular") (archivo)** Grep tiene una función compatible con Perl Regular Expressions, lo que permite un mejor filtrado (<https://cheatography.com/davechild/cheat-sheets/regular-expressions/>) `grep -P "\d{3} \d{3} \d{3}" telefonos.txt`
- **grep -E ("expresion regular") (archivo)** Grep también emplea expresiones regulares extendidas, mucho más limitadas que las anteriores `grep -E '[0-9]{3} [0-9]{3} [0-9]{3}' telefonos.txt`

## Escritura

- **cat > (nombre)** Creación directa en terminal de un archivo de texto.
- **cat (nombre1) (nombre2) > (nuevo3)** Concatena los textos de nombre1 y nombre2 en un nuevo archivo llanado nombre3
- El signo > significa output o "fuera", el output de un comando pasa a residir a un archivo. Si se usa con "cat" es solamente porque este comando concatena en la terminal.
- Puede hacerse con el echo: **echo "Buenas" > (nombre)**
- Con el comando xclip pueden depositarse las salidas de comandos al portapapeles, por ejemplo: `echo foo | xclip -selection clipboard`

## Modificación de ficheros de texto

### Cambiar caracteres

- Convertir todo a mayúsculas: **cat (archivo) | tr a-z A-Z > (archivo)** Usa el tr (translator) para modificarlo.
- El símbolo | significa que el output del primer comando va a ser el input del segundo comando. Puede hacerse tantas veces como se desee, de modo que podemos utilizar **cat (archivo) | tr a-z A-Z | tr ", " ". " > (archivo)**

### Convertir ficheros a otros formatos

- Para convertir ficheros a otros formatos puede usarse pandoc `sudo apt install pandoc`
- Si desean realizarse con el pdf-engine de weasyprint `sudo apt install weasyprint`

- **pandoc --from=(formato) --to=pdf -o (output.formato) (input.extension) --pdf-engine=weasyprint** Para convertir cualquier formato a PDF. También se puede *\*pandoc (archivo.formato) (archivo.formato) --pdf-engine=weasyprint \**
- **pandoc (archivo.formato) (archivo.formato)**

## Creación de carpetas

- **mkdir (nuevo\_nombre):** Crea una carpeta.
- (Si ingresamos **cd !\$** accederemos a la carpeta recién creada)
- **mkdir (nombre1)/(nuevo\_nombre)** Crea una carpeta dentro de una carpeta
- **mkdir -p (nombre1)/(nuevo\_nombre)/(nuevo\_nombre)** Crear varios niveles de carpetas a la vez.

## Creación de archivos

- **touch (nombre)** Crea un archivo

## Copiar archivos

- **cp (archivo) (archivo)** Copia un archivo, sobre el mismo directorio u otros.
- **cp -R (archivo)** Copia recursivamente un archivo, es decir, puede copiar carpetas.

## Mover y renombrar archivos

- **mv (archivo) (archivo)** Mueve el archivo a la ubicación. Puede cambiarse el nombre. Si el directorio es el mismo, lo renombra.

## Eliminar archivos

- **rm (archivo)** Elimina el archivo seleccionado
- **rm -R (archivo)** Elimina el archivo seleccionado recursivamente, para carpetas
- **rm -R -f** Fuerza la eliminación, evitando preguntas .

## Eliminar directorios

**rmdir (archivo)** Elimina el directorio indicado.

## Comprimir archivos



- **\*zip (nombre.zip) (archivo) . . . \*** Permite comprimir un archivo/s o carpeta
- **unzip (archivo) ...** descomprime

## Ver el historial de comandos

---

- **history** Visualiza el historial de comandos
- **history | grep (consulta)** Si se usa el signo "|" remitimos el output a la entrada del grep, con la cual podremos filtrar los comandos, por ejemplo, según su
- **!(numero)** realiza el comando indicado en el **history**

## Crear alias

---

- Los alias son formas de abreviar o crear nuevos comandos, se introducen en **/.bashrc** en los sistemas que usen bash
- Debe insertarse en el documento la siguiente estructura: **alias nombre\_alias='comando'**
- Hecho el cambio, debe utilizarse **source ~/.bashrc** para actualizar la fuente.
- Si se ejecuta **alias nombre\_alias='comando'** en la terminal, aplicará un cambio temporal que permitirá utilizar el alias solo durante la sesión actual

## Atajos

---

- **Flechas direccionales (arriba/abajo)** Ver comandos anteriores o posteriores
- **Tabulador** Autocompleta los comandos
- **CTRL+A** Moverse al principio de la línea <-
- **CTRL+E** Moverse al final de la línea ->
- **CTRL+U** Borra la línea
- **CTRL+K** Borra a partir del cursor
- **CTRL+R** Abre un buscador para comandos anteriores
- **CTRL+L** Limpia la terminal (**clear**)
- **!find** realiza la última búsqueda hecha con find
- **!(numero)** realiza el comando indicado en el **history**

## Redes

---

- **ping (web)** Consulta el PING de una web

- **wget (web)** Descarga el fichero asociado a una web
- **wget -O (nombre\_archivo) (web)** Descarga el fichero asociado a una web sobre el fichero nombre\_archivo.
- **hostname** Informa sobre el dominio del equipo
- **hostname -i** Informa sobre la IP del equipo
- **nslookup (dirección)** Informa sobre la IP de una web

## Sistema

---

- **top** Permite visualizar los procesos que se están ejecutando en el sistema, pero **htop** ofrece una mejor interfaz.
- **kill (id\_proceso)** Detiene el proceso específico asociado a la ID indicada en **top**
- **uname -a** Muestra toda la información relacionada con el sistema, el comando **neofetch** ofrece una mejor interfaz
- La carpeta **/etc/os-release** (**cat /etc/os-release**) contiene toda la información relacionada con el sistema operativo.
- **lscpu** Muestra información sobre la CPU
- **free** Muestra la memoria libre
- **vmstat** Muestra la memoria virtual
- **lsof** Muestra los archivos abiertos, junto con su ubicación
- **lsof -u (usuario)** ... por un determinado usuario
- **ps aux** Procesos actualmente reproduciéndose en el equipo

<https://youtu.be/iwoIPf6kN-k?t=6184>

## Operadores

---

- **;** para separar múltiples comandos en una sola línea, se ejecutan secuencialmente.
- **&&** ejecuta un comando únicamente cuando el proceso del anterior ya ha finalizado y no arroja error
- **&** para ejecutar simultáneamente dos comandos, por ejemplo: **ping google.es & ping yahoo.com**, el primer proceso se ejecuta internamente, por lo que no puede detenerse si antes no se le ejecuta un **kill**

- **||** significa "else", si la ejecución del primer comando falla, se ejecuta el segundo
- **!** significa "not", por ejemplo al eliminar los archivos de una carpeta: `rm -Rf !(archivo) -` entre paréntesis -
- **|** Redirige el output de un comando para que el otro lo trabaje como input
- **>** Redirige el output de un comando y lo deposita, por ejemplo, en un archivo de texto, borrándolo y sobrescribiéndolo totalmente
- **>>** Redirige el output de un comando y lo añade en un archivo de texto, pero no lo sobrescribe totalmente
- **{comando; comando}** Operador de combinación, para ejecutar varios comandos a la vez: por ejemplo: `echo "hey" && { echo "que pasa"; echo "jaja" }`