

Glosario de comandos

| COMANDO | DESCRIPCIÓN | OPCIONES | EJEMPLO |
|---------|---|---|---|
| ls | <p>Este comando se usa para mostrar el contenido de una carpeta. Tras la orden escribimos la ruta de la carpeta que queramos mostrar.</p> <p>Vale una ruta absoluta o una relativa.</p> <p>Además podemos pasar una expresión regular para filtrar dentro de la carpeta</p> | <p>--help</p> <p>-l</p> <p>-R</p> <p>-a</p> | <p>ls /home/alumnoM</p> <p>ls carpeta1</p> <p>ls *.txt</p> <p>ls carpeta1/*.txt</p> |
| cd | <p>Se usa para movernos a otra carpeta. Recordad que al usar la terminal siempre estáis en una carpeta dada. Con este comando la cambiáis.</p> <p>Acepta una argumento y es la ruta de la carpeta donde queréis ir.</p> <p>“..” es siempre la ruta de la carpeta padre de una dada.</p> | <p>--help</p> | <p>cd /home</p> <p>cd /etc</p> <p>cd carpeta1</p> <p>cd ..</p> <p>cd ../carpeta1</p> <p>cd ../../..</p> |
| touch | <p>Crea un archivo.</p> <p>El primer parámetro es la ruta del archivo que queremos crear.</p> <p>Si solamente escribimos un nombre, lo creará en la carpeta donde ejecutemos el comando.</p> | | <p>touch</p> <p>carpeta1/archivo.txt</p> <p>touch archivo.txt</p> |
| cat | <p>Abre el archivo que pasemos como primer parámetro y muestra su contenido en la</p> | | |

| | | | |
|------|--|--|--|
| | terminal. | | |
| grep | Busca una cadena de caracteres dentro de un texto. Hay dos formas de usarlo, tomando el texto como salida de otro comando usando o abriendo el archivo pasado como segundo parámetro | | ls carpeta1 grep hola grep hola archivo1.txt |
| nano | Abre el editor de texto Nano en la terminal. El primer parámetro es la ruta del archivo que queremos abrir. Si no existe, lo crea | | nano archivo.txt |
| wc | Cuenta las líneas, palabras y caracteres del archivo pasado como parámetro. También se puede usar redirigiendo la salida de otro comando a la entrada de wc. | -l líneas -w palabras -c caracteres | wc archivo.txt ls -l wc -l |
| pwd | Devuelve la ruta en la que nos encontramos en la terminal. No tiene parámetros. | | |
| cp | Copia los archivos/carpetas pasados como parámetros en la carpeta pasada como último parámetro | -r para carpetas, que se haga recursivamente | Cp manual.txt guia.txt docs copia manual.txt y guia.txt dentro de la carpeta docs |
| mv | mueve los archivos/carpetas pasados como parámetros en la carpeta pasada como último parámetro | -r para carpetas, que se haga recursivamente | mv manual.txt guia.txt docs mueve manual.txt y guia.txt dentro de la carpeta docs |
| tree | Es como ls -R pero con mejor visual. Lista la estructura de la carpeta pasada como primer parámetro. Si no hay | | |

| | | | |
|------|---|--|--|
| | parámetros, la carpeta actual | | |
| find | Busca los archivos que cumplan con los filtros pasados como opciones dentro de la carpeta pasada como primer parámetro. Si no se le pasa carpeta, es la actual. | | |

Redirección de Salidas de comandos

| | | |
|----|--|---|
| > | Sirve para redirigir la salida del comando a la izquierda del símbolo a que se almacene en un archivo en vez de mostrarse por pantalla | Ls . > archivos.txt almacena todo lo que haya en la carpeta actual en el archivo archivos.txt |
| | Sirve para redirigir la salida del comando a la izquierda como entrada del comando a la derecha | Ls -l wc -l cuenta las líneas que ha devuelto la orden ls -l, o sea, cuenta los archivos/carpetas que haya en la carpeta actual |
| 2> | Es igual que > , pero en este caso, solo almacena en el archivo los errores que haya dado el comando | Find . -name ".txt" 2> errores.txt almacena en errores.txt los errores que se haya encontrado el comando find durante su ejecución y muestra por pantalla los archivos que se haya encontrado. |
| >> | Similar a > pero en vez de sobrescribir, añade el contenido al final del archivo. | Cat archivo1.txt >> archivo2.txt En archivo2.txt se queda lo que tuviera, y debajo, el contenido de archivo1.txt |

Metacaracteres y patrones de archivo

En Linux, los **metacaracteres** permiten realizar búsquedas y operaciones más eficientes con archivos y directorios. Estos caracteres son utilizados en conjunto con comandos como **ls**, **cp**, **mv**, y **rm** para manejar múltiples archivos o directorios sin tener que escribir cada nombre individualmente.

Principales metacaracteres

- ***** (asterisco): Representa **cero o más caracteres**. Se utiliza para coincidir con cualquier número de caracteres en un nombre de archivo o directorio.
 - Ejemplo: `ls *.txt` listará todos los archivos que terminen con la extensión `.txt` en el directorio actual.
 - Ejemplo: `rm *` eliminaría todos los archivos del directorio actual (ten cuidado con este comando).
- **?** (interrogación): Representa **un único carácter**. Se usa para coincidir con cualquier carácter en una posición específica dentro del nombre de un archivo o directorio.
 - Ejemplo: `ls archivo?.txt` mostrará archivos que tengan un nombre como `archivo1.txt`, `archivo2.txt`, pero no `archivo10.txt`.
- **[]** (corchetes): Permiten especificar un **rango de caracteres**. Solo coinciden con uno de los caracteres dentro de los corchetes.
 - Ejemplo: `ls archivo[1-3].txt` listará `archivo1.txt`, `archivo2.txt`, y `archivo3.txt`, pero no `archivo4.txt`.
- **{ }** (llaves): Se usa para crear **listas de opciones**. Esto permite generar múltiples coincidencias en una sola expresión.
 - Ejemplo: `ls archivo{1,2,3}.txt` listará `archivo1.txt`, `archivo2.txt`, y `archivo3.txt`.

Ejemplos de uso

1. Buscar archivos por extensión:

- `ls *.jpg`: Listará todos los archivos con la extensión `.jpg`.

2. Eliminar archivos con nombres similares:

- `rm temp?.log`: Eliminará archivos como `temp1.log`, `temp2.log`, pero no `temp10.log`.

3. Copiar archivos con nombre en un rango específico:

- `cp archivo[1-5].txt /ruta/destino`: Copiará archivos como `archivo1.txt`, `archivo2.txt`, hasta `archivo5.txt` al destino.

4. Mover varios archivos con nombres definidos:

- `mv archivo{1,2,3}.txt /ruta/destino`: Moverá `archivo1.txt`, `archivo2.txt`, y `archivo3.txt` al destino.