

Terminal

Christian Gimenez

18 may 2024



Este trabajo se encuentra bajo la licencia Creative Commons Attribution-ShareAlike 4.0 International.

Información de la licencia: <http://creativecommons.org/licenses/by-sa/4.0/>

El código fuente se encuentra en [el proyecto Emacs-Spellbook](#) (ver [archivos Org-mode](#)).

Índice general

1	Acerca de este documento	3
2	Introducción	3
3	Acerca de los directorios	4
4	Anatomía de una terminal	5
5	Manuales y ayuda	6
6	Manipular archivos	8
7	Ver y editar archivos	9
7.1	Nano	10
8	Gestionar procesos	10
9	Saber dónde estamos	13
10	¿Más?	13

1 Acerca de este documento

Esta obra es una referencia de los conceptos básicos, programas y comandos más utilizados de la terminal. Si bien, puede leerse para aprender y estudiar, es importante practicar los comandos para comprender mejor y afianzar los contenidos.

Por ello, se recomienda utilizar un tutorial en el que paso a paso se practiquen los comandos, teniendo cerca este documento para comprender qué tarea se está realizando. También, se recomienda (en tanto no sea abrumador) intentar utilizar la terminal para tareas sencillas que habitualmente haríamos en la interfaz gráfica (por ejemplo: mover archivos, cambiarles de nombre, crear carpetas, etc.).

2 Introducción

1. Las terminales utilizan un programa llamado «shell» que interpreta comandos.

Existen: sh, bash, fish, ksh, zsh, csh, entre otras.

2. Cada shell tiene una sintaxis. Pero, para empezar a usarlas, es útil recordar la siguiente regla:

```
COMANDO-O-PROGRAMA ARGUMENTO1 ARGUMENTO2 ARGUMENTO3 . . .
```

En otras palabras, lo primero que se escribe es el comando y luego los argumentos separados con espacios.

3. Muchas shell poseen un directorio actual.

Los comandos pueden trabajar en relación a este directorio actual. Por ejemplo, se puede ejecutar `ls` para que muestre los archivos, pero si no se le indica cuál directorio usar, utilizará el actual.

4. Muchos shell proveen autocompletado y ofrecen sugerencias. Esta característica se activa presionando TAB una vez para autocompletar, y dos veces para ver sugerencias.

También, ofrecen historial de comandos y otras características que pueden ayudarte a encontrar lo que buscás y ahorrar mucho tipeo.

5. La mayoría de los programas reconocen al atajo `Control C` para cancelar de forma brusca y salir inmediatamente al shell. Si está ejecutando un programa, y desea salir de él, puede utilizarlo.

 Detener un programa con `Control C` implica interrumpir lo que estaba haciendo y salir inmediatamente. Esto puede traer consecuencias no deseadas.

- (a) ej.: si está copiando un archivo muy grande con el comando `cp gran-archivo.iso Documentos/`, puede utilizar `Control C` para terminar repentinamente la ejecución. Sin embargo, debe asegurarse si se copió algún contenido y de borrar la copia incompleta si hubiese una. En el ejemplo, debe revisar que el archivo «Documentos/gran-archivo.iso» exista y eliminarlo puesto que no es el mismo.

3 Acerca de los directorios

1. Los directorios se separan con «/» en sistemas Unix, GNU/Linux, BSD y similares.

- (a) ej.: `/home/juana/miarchivo.txt` indica el directorio que se encuentra en la raíz («/»), allí dentro de «home», luego dentro del directorio «juana», y dentro de este, el archivo «miarchivo.txt».

2. El directorio «~» es una abreviatura para acceder rápidamente a tu directorio personal (común mente denominado «home»).

Por ejemplo, si sos la usuaria «juana» en el sistema, `~/miarchivo.txt` significa lo mismo que «`/home/juana/miarchivo.txt`», que es el archivo «miarchivo.txt» dentro de su directorio personal. Informalmente decimos que «miarchivo.txt se encuentra dentro del home de juana».

3. El directorio «.» y «..» significan el directorio actual y el padre respectivamente.

Por ejemplo: El siguiente comando copia «miarchivo.txt» que está en el directorio «Documentos» al directorio actual.

```
cp Documentos/miarchivo.txt .
```

4. Al principio, no utilizar nombres complicados y con signos.

⚠ Totalmente evitar nombres de archivos con asteriscos (*), tilde (~) y barras inclinadas (/). Si se desea, utilizar el guión (-) o guión bajo (_) en vez de espacio.

5. Si los nombres de directorios y archivos tienen espacios, encerrarlos con comillas simples.

Por ejemplo: El siguiente comando no copia «mi archivo.txt» al directorio «Sumamente Importantes», sino que intenta copiar los archivos «mi», «archivo.txt», «Documentos» y «Sumamente» al directorio «Importantes», pudiendo fallar si no los encuentra todos.

```
cp mi archivo.txt Documentos Nuevos/Sumamente Importantes
```

En cambio, el siguiente sí copia «mi archivo.txt» a «Sumamente Importantes».

```
cp 'mi archivo.txt' 'Mis Documentos/Sumamente Importantes'
```

👉 Este es un error muy habitual para quienes comienzan. Por suerte, muchos shell tienen autocompletado: a medida que escribimos, si presionamos la tecla TAB, el shell intentará completar el nombre del archivo o directorio si es la única alternativa.

4 Anatomía de una terminal

En la [1](#) se muestra una captura de pantalla de una terminal con descripciones. Se pueden observar los siguientes ítems:

El programa emulador Es el programa que muestra en pantalla a la terminal. Un emulador contiene a un shell.

Prompt El signo «\$» o «#» y lo que está antes se denomina «prompt». Indica el estado de la terminal y el directorio actual.

Salidas de los programas Luego de ejecutar un programa, se observa sus salidas. En este caso, se señalan las del programa emacs y las del comando `cat notas.txt`.

```

Programa emulador de terminal
~/mi-directorio
~ $ mkdir mi-directorio
~ $ cd mi-directorio/
~/mi-directorio $ mkdir importantes
~/mi-directorio $ emacs notas.txt
Gtk-Message: 19:47:34.911: Failed to load module "appmenu-gtk-module"
~/mi-directorio $ cat notas.txt
Mis notas ...
Las escribo aqui.
~/mi-directorio $ cp notas.txt importantes/
~/mi-directorio $ mv -v notas.txt mis-notas.txt
renamed 'notas.txt' -> 'mis-notas.txt'
~/mi-directorio $ ls -lh
total 8,0K
drwxr-xr-x 2 poo poo 4,0K may 18 19:48 importantes/
-rw-r--r-- 1 poo poo 33 may 18 19:48 mis-notas.txt
~/mi-directorio $ ls
importantes/ mis-notas.txt
~/mi-directorio $ file mis-notas.txt
mis-notas.txt: Unicode text, UTF-8 text
~/mi-directorio $ tree
.
├── importantes
│   └── notas.txt
└── mis-notas.txt

2 directories, 2 files
~/mi-directorio $

```

Figura 1: Descripción de una terminal.

La tilde o virgulilla («~») en el **prompt** Para ahorrar espacio, usualmente se muestra esta abreviatura en vez de /home/USUARIO.

Una comprobación después de utilizar mv Como es recomendado, se utiliza `ls -lh` para asegurarse si la instrucción `mv` anterior realizó la tarea que esperábamos.

El tipo de archivo Se utiliza el programa `file` para ver qué tipo de archivo es «mis-notas.txt». En ocasiones, se muestran más detalles del contenido del archivo, en este caso, la codificación.

El punto («.») Se utiliza como referencia al directorio actual. En este ejemplo se observa la salida del comando `tree`, que muestra el contenido del directorio actual y sus sub-directorios.

5 Manuales y ayuda

Los siguientes comandos son muy útiles para buscar manuales completos o ayuda de un comando en especial. Se encuentran disponibles en la mayoría de los sistemas GNU/Linux.

Comando	Descripción
<code>info</code>	Listado de manuales, con hipervínculos. También se pueden encontrar en Internet.
<code>info MANUAL</code>	Si conocemos el título del manual, se puede abrir directamente con este comando.
<code>man PROGRAMA</code>	Manpages, son páginas o manuales que se elaboran para un programa en particular.
<code>PROGRAMA --help</code>	La mayoría de los programas muestran una ayuda al utilizar el parámetro «help».

Después de ejecutar `info`, se navega los manuales con las siguientes teclas:

Teclas	Descripción
<code>←, →, ↑, ↓</code>	Desplazar el cursor
<code>Enter</code>	Entrar a un vínculo
<code>q</code>	Salir

Manuales recomendados:

<code>info coreutils</code>	Documentación de todos los programas básicos y habituales
<code>info nano</code>	Documentación del editor de texto nano
<code>info bash</code>	Documentación del shell bash

6 Manipular archivos

<code>pwd</code>	Mostrar el directorio actual
<code>cd DIRECTORIO</code>	Cambiar directorio actual por DIRECTORIO
<code>cd</code>	Cambiar directorio actual por el home
<code>ls</code>	Listar los archivos del directorio actual
<code>ls -lh</code>	Listar archivos con más detalles
<code>ls DIRECTORIO</code>	Listar los archivos de DIRECTORIO
<code>ls -lh DIRECTORIO</code>	Listar DIRECTORIO con más detalles
<code>file ARCHIVO</code>	Muestra el tipo de archivo
<code>tree</code>	Listar los archivos en forma de árbol
<code>tree DIRECTORIO</code>	Listar DIRECTORIO en forma de árbol
<code>mv ORIGEN DESTINO</code>	Mover archivos desde ORIGEN a DESTINO También, renombra archivos si coinciden los directorios.
<code>mv -v ORIGEN DESTINO</code>	Análogo al anterior, pero muestra qué hace en pantalla. Útil para asegurarse.
<code>cp ORIGEN DESTINO</code>	Copiar el archivo ORIGEN a DESTINO
<code>rm ARCHIVO</code>	Eliminar un archivo (¡no se recuperan!)
<code>mkdir DIRECTORIO</code>	Crear un directorio
<code>rmdir DIRECTORIO</code>	Eliminar un directorio

💡 Después de usar `cp`, `mv` o `rm`, se recomienda **corroborar si el programa no reporta error y si el archivo se encuentra copiado/movido/eliminado correctamente con `ls` o `file`**.

👉 `cp` y `rm` solo funcionan con archivos, no con directorios.

💡 Si al principio, `mv` le resulta complicado de aprender, puede copiar el archivo ORIGEN a DESTINO, y después eliminar el archivo ORIGEN:

```
cp ORIGEN DESTINO
rm ORIGEN
```

También, puede utilizar `mv -v ORIGEN DESTINO` para que el comando reporte qué acciones hace. Recuerde que «-v» proviene de «-verbose» (verborreico).

👉 `mv` acepta mover directorios y archivos.

⚠️ `mv` **sobreescribe el destino si es un archivo existente** y el origen también es un archivo. Ver ejemplos más abajo.

⚠️ En la terminal **no existe la papelera de reciclaje**. Si se borra un archivo con `rm` o un directorio con `rmdir`, no se podrán recuperar.

👉 `rmdir` no borrará el directorio si contiene un archivo.

El comando `mv` renombra archivos cuando el directorio coincide o le indicamos como archivo destino uno que no existe. Por ejemplo:

El siguiente comando renombra «mi archivo.txt» a «mi archivo2.txt». Si «mi archivo2.txt» ya existe, **lo sobreescribe**.

```
mv 'mi archivo.txt' 'mi archivo2.txt'
```

El siguiente comando mueve «mi archivo.txt» a «Documentos/» y le cambia de nombre a «mi archivo2.txt». Si «Documentos/mi archivo2.txt» existe, **lo sobreescribe**. Si «Documentos/» no existe, reportará un error y no realizará ninguna acción.

```
mv 'mi archivo.txt' 'Documentos/mi archivo2.txt'
```

7 Ver y editar archivos

<code>cat ARCHIVO</code>	Muestra el contenido de un archivo
--------------------------	------------------------------------

– Editores –

<code>nano ARCHIVO</code>	Un editor sencillo.
---------------------------	---------------------

<code>vim ARCHIVO</code>	
--------------------------	--

<code>emacs ARCHIVO</code>	
----------------------------	--

También se puede usar `gedit`, `kate`, `kwrite`, `pluma` u otros utilizados en entornos gráficos. Usualmente, se ejecutan de la misma manera como los anteriores. P. ej.: para editar el archivo «mi archivo.txt» con `kate`, se debe ejecutar:

```
kate 'mi archivo.txt'
```

7.1 Nano

Nano es un editor de texto pequeño y fácil de usar. Se encuentra instalado por defecto en la mayoría de las distribuciones GNU/Linux. Su ventaja es que es rápido y posee las características habituales de muchos editores.

Para abrir nano, solo se debe ejecutar `nano` en la terminal. Si se ejecuta con un archivo como argumento, abrirá y mostrará dicho archivo.

En la [2](#) se muestra al editor abriendo el archivo `wikipedia.html`. Para ello, el archivo `wikipedia.html` se encontraba en el directorio actual de la terminal, y se ejecutó el siguiente comando:

```
nano wikipedia.html
```

En la [3](#), se aprecia en la zona inferior, la cantidad de líneas que posee el archivo, indicando con «Read 863 lines» que posee 863 líneas. Debajo de esto se presenta una leyenda de los acceso de teclado disponibles. Considere que «`^`» significa Control y «`M-`» significa Alt. De esta manera, «`^X`» significa «Control X» y al presionar dichas teclas saldría de nano (*exit* en inglés significa salir).

8 Gestionar procesos

Todos los procesos y programas en ejecución del sistema tienen un identificador llamado *Process ID* (PID). Éste es un número que se le asigna al momento de cargar e iniciar su ejecución.

También, se puede enviar señales a los procesos. La señal para indicarles que terminen correctamente es la número 15 (conocida como TERM). Usualmente, un programa cierra los archivos y se cierra ante esta señal.

Cuando esto no sucede por razones imprevistas (p. ej. por bucles infinitos o errores), puede utilizarse la señal 9, llamada KILL. Esta le indica al proceso que termine abruptamente, sin realizar ninguna limpieza ni cerrar archivos.

```
nano wikipedia.html ~
GNU nano 7.2 wikipedia.html
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
<meta charset="utf-8">
<title>Wikipedia</title>
<meta name="description" content="Wikipedia is a free online encyclopedia, created and edited by vol>
<script>
document.documentElement.className = document.documentElement.className.replace( /(^|\s)no-js(\s|$)/>
</script>
<meta name="viewport" content="initial-scale=1,user-scalable=yes">
<link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png">
<link rel="shortcut icon" href="/static/favicon/wikipedia.ico">
<link rel="license" href="//creativecommons.org/licenses/by-sa/4.0/">
<style>
.sprite{background-image:linear-gradient(transparent,transparent),url(portal/wikipedia.org/assets/im>
</style>
<style>
html{font-family:sans-serif;-ms-text-size-adjust:100%;-webkit-text-size-adjust:100%;font-size:62.5%}>
</style>
<link rel="preconnect" href="//upload.wikimedia.org">
<link rel="me" href="https://wikis.world/@wikipedia">
<meta property="og:url" content>
<meta property="og:title" content="Wikipedia, the free encyclopedia">
<meta property="og:type" content="website">
<meta property="og:description" content="Wikipedia is a free online encyclopedia, created and edited>
<meta property="og:image" content="https://upload.wikimedia.org/wikipedia/en/thumb/8/80/Wikipedia-lo>
</head>
<body id="www-wikipedia-org">
<main>
<div class="central-textlogo">

<h1 class="central-textlogo-wrapper">
<span class="central-textlogo__image sprite svg-Wikipedia_wordmark">
Wikipedia
</span>
<strong class="jsl10n localized-slogan" data-jsl10n="portal.slogan">The Free Encyclopedia</strong>
[ Read 863 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
```

Figura 2: Editor de texto Nano.

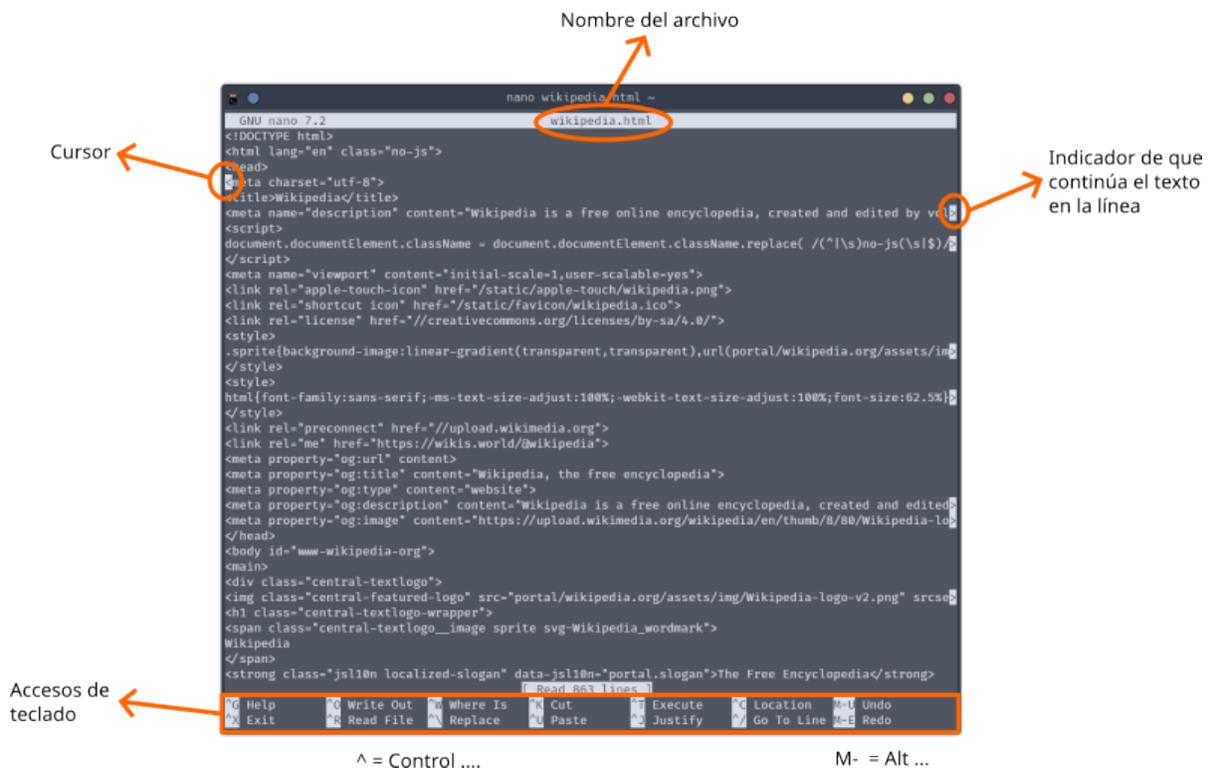


Figura 3: Descripción de la interfaz de Nano.

<code>ps -A</code>	Mostrar todos los procesos del sistema
<code>pstree</code>	Mostrar los procesos en forma de árbol
<code>free -h</code>	Muestra el estado de la memoria del sistema
<code>kill -TERM PID</code>	Termina un proceso.
<code>kill -15 PID</code>	Termina un proceso (15 = TERM)
<code>kill -KILL PID</code>	«Mata» un proceso.
<code>kill -9 PID</code>	«Mata» un proceso (9 = KILL)

💡 Se puede utilizar el siguiente comando busca una palabra que contiene el programa en ejecución. Es útil para obtener el PID de un proceso.

`ps -Af | grep PALABRA`

💡 El programa `top` puede usarse para ver el estado completo de los procesos. Es un programa interactivo, muy configurable, que muestra continuamente la situación del CPU, memoria y Entrada/Salida. Se recomienda aprender a usarlo. Recuerde que con la tecla «q» sale del programa.

9 Saber dónde estamos

<code>pwd</code>	Mostrar el directorio actual
<code>whoami</code>	Mostrar el usuario actual
<code>uname -a</code>	Mostrar datos del sistema
<code>uptime</code>	Cuánto tiempo el sistema estuvo encendido
<code>date</code>	Mostrar la fecha y hora actual

10 ¿Más?

Se puede:

- Ejecutar varios programas a la vez.
 - Escuchar música y seguir usando la terminal.
- Utilizar varios terminales (multiplexar con `screen` o `tmux`).
- Acceder a terminales remotas.
- Manipular el sistema completo: procesos, ver uso del CPU y memoria, entradas y salidas actuales, etc.