

Introducción a la terminal de Linux

DigitalHouse>



**Certified Tech
Developer**

The Ultimate Degree

Índice

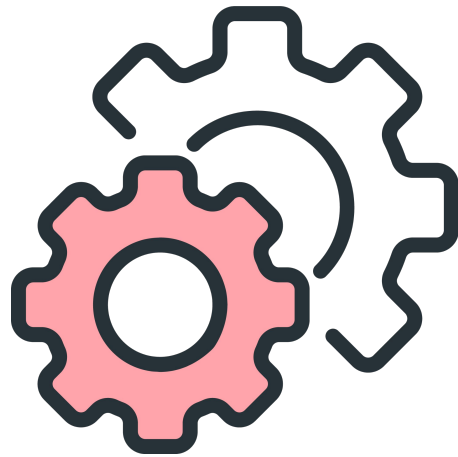
1. [La consola](#)
2. [Diferentes tipos de shell](#)
3. [Ejecución de la consola](#)
4. [Elevación de privilegios](#)

1 | La consola

La consola de Linux

La interfaz de línea de comandos, o CLI —por sus siglas en inglés *command-line interface*—, es un método de comunicación entre usuario y máquina que acepta instrucciones del usuario a través de líneas de texto (siguiendo unas determinadas reglas de sintaxis que puedan ser interpretadas por el sistema operativo).

La herramienta que posibilita la función de interfaz de usuario se la denomina shell. Aplicado en el ámbito de la interfaz de línea de comandos, estaríamos hablando de una **shell CLI o intérprete de comandos**.



2 | Diferentes tipos de shell

Tipos de shell

En Linux tenemos una multitud de shells o intérpretes diferentes. El más conocido de todos probablemente es Bash, debido a que es el que suele venir por defecto en la gran mayoría de distribuciones GNU/Linux, pero también destacan otros como Bourne Shell (sh), Korn Shell (ksh) o C Shell (csh), los cuales vamos a conocer.



```

( Today is National Existential Ennui )
( Awareness Day. )
-----
o
o
      .-.
     |o o|
     | 7 |
    /   \
   /       \
  /         \
 /           \
/             \
\             /
 \           /
  \         /
   \       /
    \     /
     \|_/

```

```
metalk1000@taka ~$ man pv
metalk1000@taka ~$ echo "This is some text"
This is some text
metalk1000@taka ~$ echo "This is some text"
This is some text
metalk1000@taka ~$ echo "This is some text"|pv -qL 10
This is some text
metalk1000@taka ~$ echo "This is some text"|pv -qL 10
This is some text
metalk1000@taka ~$ echo "This is some text"|pv -qL 20
This is some text
metalk1000@taka ~$ echo "This is some text"||pv -qL 100
This is some text
metalk1000@taka ~$ █
```

Bourne Shell

Lleva el nombre de su creador en los Laboratorios Bell, Steve Bourne. Fue la primera shell utilizada para el sistema operativo Unix y ha superado en gran parte la funcionalidad de muchas de las shells más recientes. Todas las versiones de Linux Unix permiten a los usuarios cambiar a la original Bourne Shell, conocida simplemente como "sh", si así lo desean. Sin embargo, hay que tener en cuenta que al hacerlo, se renuncia a funcionalidades como el completado de nombres de archivo y el historial de comandos que los depósitos posteriores han añadido.

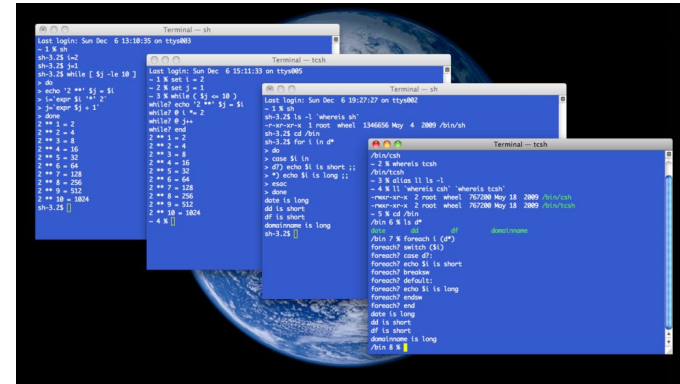


```
Terminal
-rwxr-xr-x 1 bin 18296 Jun 8 1979 fsck
-rwxr-xr-x 1 bin 1458 Jun 8 1979 getty
-rw-r--r-- 1 root 49 Jun 8 1979 group
-rwxr-xr-x 1 bin 2482 Jun 8 1979 init
-rwxr-xr-x 1 bin 8484 Jun 8 1979 mkfs
-rwxr-xr-x 1 bin 3642 Jun 8 1979 mknod
-rwxr-xr-x 1 bin 3976 Jun 8 1979 mount
-rw-r--r-- 1 root 141 Jun 8 1979 passwd
-rw-r--r-- 1 bin 366 Jun 8 1979 rc
-rw-r--r-- 1 bin 266 Jun 8 1979 ttys
-rwxr-xr-x 1 bin 3794 Jun 8 1979 umount
-rwxr-xr-x 1 bin 634 Jun 8 1979 update
-rw-r--r-- 1 bin 40 Sep 22 05:49 utmp
-rwxr-xr-x 1 root 4520 Jun 8 1979 wall
# ls -l /*unix*
-rwxr-xr-x 1 sys 53302 Jun 8 1979 /hptunix
-rwxr-xr-x 1 sys 52850 Jun 8 1979 /hptmunix
-rwxr-xr-x 1 root 50990 Jun 8 1979 /rkunix
-rwxr-xr-x 1 root 51982 Jun 8 1979 /rl2unix
-rwxr-xr-x 1 sys 51790 Jun 8 1979 /rphtunix
-rwxr-xr-x 1 sys 51274 Jun 8 1979 /rptmunix
# ls -l /bin/sh
-rwxr-xr-x 1 bin 17310 Jun 8 1979 /bin/sh
#
```

C/TC Shell

El C Shell fue desarrollado posteriormente al Bourne Shell y está pensado en facilitar el control del sistema al programador en lenguaje C. La razón de esto es que su sintaxis, como vamos a apreciar, es muy similar a la de este lenguaje.

Conocido popularmente también como **csh**, está presente en otros SO, por ejemplo, en Mac OS. Posee una evolución, conocida como **tcsh** que incorpora funcionalidades avanzadas y mayores atajos de teclado.



Korn Shell

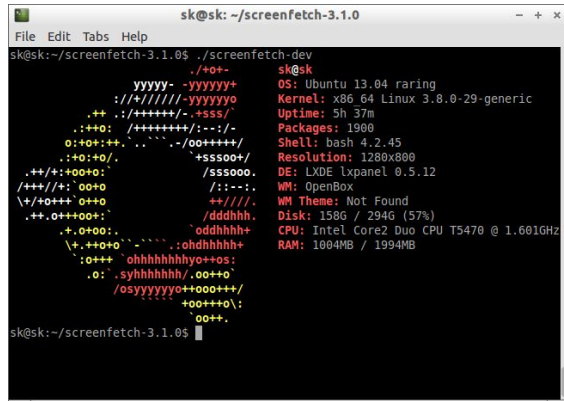
Esta también fue escrita por un programador en los Laboratorios Bell, David Korn. Intenta combinar las características de la C Shell, TC Shell y Bourne Shell en un solo paquete. También incluye la capacidad para crear nuevos comandos de shell para los desarrolladores cuando surja la necesidad.

```
QEMU
# uname -a
OpenBSD berkeley.my.domain 5.3 GENERIC#53 amd64
# pwd
/tmp
# cd ~
# ls -l
total 2112
-rw-r--r--  1 root  wheel   112 Jun 20  2013 .Xauthority
-rw-r--r--  1 root  wheel    22 Mar 13  2013 .Xdefaults
-rw-r--r--  1 root  wheel   578 Mar 13  2013 .cshrc
drwxr-xr-x  3 root  wheel   512 Jun 24  2013 .gem
-rw-r--r--  1 root  wheel   125 Mar 13  2013 .klogin
-rw-r--r--  1 root  wheel   328 Mar 13  2013 .login
-rw-r--r--  1 root  wheel   552 Jun 24  2013 .profile
-rw-r--r--  1 root  wheel    63 Jun 20  2013 .serverauth.18088
-rw-r--r--  1 root  wheel 1044692 Jun 28  2013 mbox
# whoami
root
# ls -l $(which vi)
-r-xr-xr-x  3 root   bin  361496 Mar 13  2013 /usr/bin/vi
# echo $SHELL
/bin/ksh
#
```



Posee funciones avanzadas para manejar archivos de comandos que la colocan a la par de lenguajes de programación especializados, como AWK y Perl.

Bourne-Again Shell (BASH)



La Bourne-Again Shell es una versión actualizada de la Bourne Shell original. Es una shell utilizada ampliamente en la comunidad de código abierto.

Su sintaxis es similar a la utilizada por la Bourne Shell, incorporando funcionalidades más avanzadas que se encuentran en las shells C, TC y Korn.



Entre las funcionalidades adicionales que carecía Bourne, está la capacidad para completar nombres de archivos pulsando la tecla TAB, la capacidad de recordar un historial de comandos recientes y la capacidad de ejecutar múltiples programas en segundo plano a la vez.

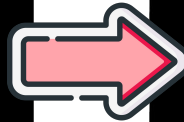
3 | Ejecución de la consola

Consola de Linux: Ejecución en inicio

Si bien cada distribución de Linux tiene su manera particular de acceder a la consola, cuando el SO se inicia en los niveles 1, 2, 3 y 4 nos llevará por defecto a la consola.

```
CentOS Linux (3.10.0-1160.6.1.el7.x86_64) ? (Core)
CentOS Linux (0-rescue-ff31bca5aa6fe84a9c27749f5242b30b) ? (Core)

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

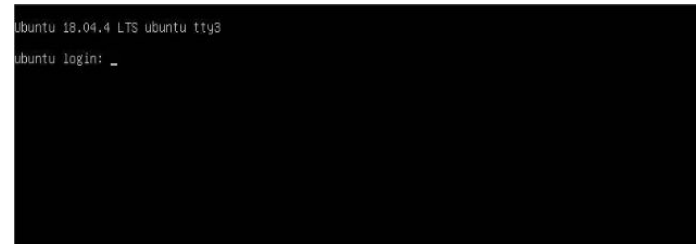
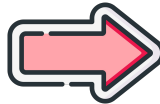


```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.6.1.el7.x86_64 on an x86_64
localhost login: _
```

Consola de Linux: Ejecución desde GUI

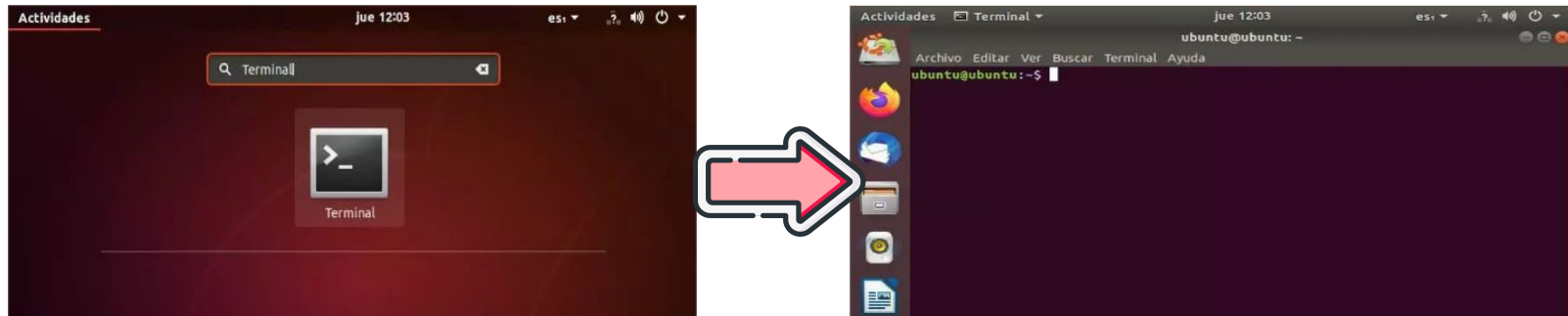
Si en cambio nuestro SO inicia en nivel 5 (con GUI), para poder utilizar la terminal tenemos diferentes opciones. Estas varían de acuerdo a la distribución instalada. En el caso de Ubuntu, tenemos dos opciones:

- La primera de ellas es lanzando un TTY, o **espacio de trabajo sin entorno gráfico**. Podemos ejecutar 7 terminales al mismo tiempo de esta forma. De la 1 a la 6, ninguna tiene interfaz gráfica. Para cambiar de TTY en Linux debemos usar el atajo de teclado Control+Alt más la tecla —de F1 al F7— del TTY que queramos ejecutar.



Consola de Linux: Ejecución desde GUI

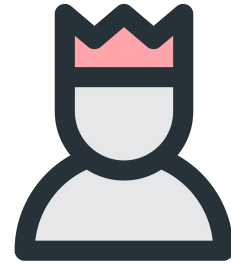
- La segunda opción es encontrar una app dedicada que se ejecuta en una ventana, dentro del panel de aplicaciones de nuestra distro. En el caso de Ubuntu, por ejemplo, podemos encontrar esta terminal dentro del cajón de programas del entorno gráfico GNOME.



4 | Elevación de privilegios

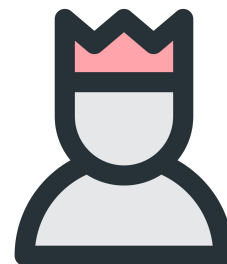
Los privilegios del superusuario root

Por lo general, los sistemas operativos contemplan el uso de solo un usuario, el cual tiene permisos de administrador. En Linux las cosas se manejan de una forma particular, se separa la cuenta de usuario común de la de superusuario y es eso lo que conocemos como root. Esta cuenta **posee todos los privilegios y permisos para realizar acciones sobre el sistema.**



Los privilegios del superusuario root

Para la ejecución de algunos comandos debemos ingresar dicho acceso (clave de root). Sin embargo, se debe tener un conocimiento sobre las acciones que se realizan, ya que **una acción realizada de manera errónea podría ocasionar daños importantes en el sistema**. El uso de instrucciones con privilegios de superusuario pueden ser sumamente útiles, pero totalmente devastadoras si desconocemos las consecuencias de su uso en el sistema. Veamos el método para elevar nuestros privilegios.



Elevando privilegios

Suponiendo que iniciamos sesión como un usuario “común”, denominado “edorio” y queremos reiniciar un servicio (cron), vamos a obtener lo siguiente:



{}

```
edorio@DESKTOP-W10:~$ service cron start
* Starting periodic command scheduler cron
cron: can't open or create /var/run/crond.pid: Permission denied
[fail]
edorio@DESKTOP-W10:~$
```

Elevando privilegios

Para evitar el error, debemos usar el comando sudo, previo al comando que queremos ejecutar. Nos pedirá la contraseña de root y se ejecutará como tal de manera satisfactoria.



{ }

```
edorio@DESKTOP-W10:~$ sudo service cron start
[sudo] password for edorio:
 * Starting periodic command scheduler cron
[ OK ]
edorio@DESKTOP-W10:~$
```

DigitalHouse>