

Comandos necesarios y esenciales

1. Comando "lsb_release"

Muestra información detallada sobre el sistema junto con el parámetro -a

```
sudo lsb_release -a
```

2. Comando "apt"

El comando APT (Advanced Package Tool), sus siglas en inglés que traducen Herramienta Avanzada de Empaquetado es un elemento de línea de comandos creado por el proyecto Debian con el objetivo de permitirle a los usuarios gestionar y administrar los paquetes de sus distribuciones de Linux Debian.

apt	->	Debian, Ubuntu, Mint.
yum	->	RedHat, CentOS, Fedora.
zip	->	SuSE, OpenSuse.

2.1. Update

```
sudo apt-get update
```

Se usa para re-sincronizar los índices de los paquetes desde las fuentes (/etc/apt/sources.list), siempre se debe ejecutar antes de upgrade o dist-upgrade

este comando se encarga de sincronizar todos los paquetes del repositorio para ver cuales han cambiado y cuales se han mantenido, no instala absolutamente nada. simplemente actualiza la lista de repositorios que se encuentra en la ruta "/etc/apt/sources.list"

```
root@debian:~# apt-get update
Des:1 http://security.debian.org/debian-security buster/updates InRelease [34,8 kB]
Obj:2 http://deb.debian.org/debian buster InRelease
Des:3 http://deb.debian.org/debian buster-updates InRelease [56,6 kB]
Des:4 http://security.debian.org/debian-security buster/updates/main Sources [296 kB]
Des:5 http://security.debian.org/debian-security buster/updates/main amd64 Packages [422 kB]
Des:6 http://security.debian.org/debian-security buster/updates/main Translation-en [228 kB]
Descargados 1.038 kB en 2s (462 kB/s)
Leyendo lista de paquetes... Hecho
```

2.2. Upgrade

```
sudo apt-get upgrade
```

instala las versiones nuevas de los paquetes y las modificaciones de los paquetes actualmente instalados, bajo ninguna circunstancia elimina paquetes ni dependencias, no es recomendable ejecutar este comando si el servidor está en producción ya que reinicia todos los servicios que estén en la lista de actualización, al ejecutar este comando el servicio de actualización pedirá una confirmación. Depende de nosotros y aceptamos o no. o también podemos usar el parámetro -y para simplificar la pregunta que no hace.

2.3. Dis-Upgrade

sudo apt-get dist-upgrade

adiciona a lo ejecutado por “upgrade” controla de manera inteligente los cambios de las dependencias, con la capacidad de resolver conflictos instalando nuevas versiones de los paquetes y eliminando las versiones anteriores de ser necesario.

3. Comando “timedatectl”

El comando **timedatectl** muestra detalles de la hora, fecha y zona horaria del servidor

```
root@prueba:/home/leonel# timedatectl
Local time: Tue 2023-01-24 21:23:52 UTC
Universal time: Tue 2023-01-24 21:23:52 UTC
RTC time: Tue 2023-01-24 21:23:51
Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

3.1. Modificar la zona horaria del servidor

Tenemos dos opciones

3.1.1. Por consola

Primero debemos obtener un listado de las zonas horarias disponibles

```
sudo timedatectl list-timezones
America/Kralendijk
America/La_Paz
America/Lima
```

Luego de tener claro a cuál zona horaria lo modificaremos, lo modificamos con el siguiente comando

sudo timedatectl set-timezone America/La_Paz

3.1.2. Por un gestor de zonas horarias

dpkg-reconfigure tzdata

es una guía más fácil para cambiar la zona horaria

4. Comando “hostnamectl”

Es un comando o una herramienta que nos ayuda a ver información o a modificar el nombre del sistema

sudo hostnamectl status

Nos muestra información detallada del sistema

sudo hostname set-hostname NuevoNombreDelSistema

con este comando cambiaremos el nombre del sistema a cualquier otro nombre

Luego de ejecutar el anterior comando se debe reiniciar el sistema

sudo reboot

Cuando el sistema vuelva a iniciar se debe modificar el fichero “/etc/hosts” para modificarlo de la siguiente forma.

```
127.0.0.1 localhost
127.0.1.1 ubu
```

5. Comando “man”

Uno de los primeros comandos que debemos conocer en Linux es **man**. Con este comando podemos conocer el uso de todos los comandos de Linux, mostrando una vista con información como nombre, sinopsis, descripción, opciones, estado de salida, valores de devolución, errores, archivos, versiones, ejemplos, etc. Por ejemplo, si quieres conocer el comando “**cd**” y sus opciones, tendrías que ejecutar el comando **man cd** para conocer su descripción y uso.

man {comando/nombreHerramienta}

6. Comando “cd”

Otros de los comandos más útiles en Linux es el comando **cd** o de cambio de directorio (Change Directory). Con la ayuda de este comando, podemos navegar por todos nuestros directorios en nuestro sistema. Las opciones que nos permite serían:

Cambiar del directorio actual a un nuevo directorio

Cambiar directorio usando una ruta absoluta

Cambiar directorio usando la ruta relativa

Cambiar al directorio de inicio

Cambiar al directorio anterior

Cambiar al directorio principal

Cambiar al directorio raíz

Cambiar al directorio de inicio de otro usuario

Cambiar a directorio con espacios

Cambiar hasta múltiples subdirectorios

La sintaxis del comando **cd** es la siguiente:

cd ..	->	para ir a un directorio hacia arriba
cd /	->	para ir al directorio raíz
cd -	->	para ir al directorio anterior

Como nota al margen, el Shell de Linux distingue entre mayúsculas y minúsculas. Por lo tanto, debes escribir el nombre del directorio de forma exacta.

7. Comando “touch”

Este comando se utiliza para crear un archivo vacío con un nombre específico. Por ejemplo, “touch archivo1.txt” creará un archivo vacío llamado “archivo1.txt” en el directorio actual.

sudo touch nombreArchivo.extencion

8. Comando “cat”

cat es uno de los más utilizados en Linux. El nombre del comando cat viene de “concatenate”, su funcionalidad para concatenar archivos o unir, sumar. Puede leer, concatenar archivos, combinarlos y escribir contenidos de archivos en una salida estándar.

Este comando se utiliza para ver el contenido de un archivo de texto. Por ejemplo, "cat archivo1.txt" mostrará el contenido del archivo "archivo1.txt" en la consola.

También se utiliza para copiar contenido de un archivo a otro, mostrar el número de línea o mostrar \$ al final de la línea.

```
sudo cat archivo.txt
```

9. Comando “ls”

El comando ls se usa para ver el contenido de un directorio. Por defecto, este comando mostrará el contenido de tu directorio de trabajo actual.

Si deseas ver el contenido de otros directorios, escribe ls y luego la ruta del directorio. Por ejemplo, ingresa ls/home/nombredeusuario/Documentos para ver el contenido de Documentos.

Hay variaciones que puedes usar con el comando ls:

ls -R también listará todos los archivos en los subdirectorios

ls -a mostrará los archivos ocultos

ls -al listará los archivos y directorios con información detallada como los permisos, el tamaño, el propietario, etc.

```
ls -lah
```

10. Comando “vim”

vim es un editor de texto de terminal gratuito y de código abierto. Puedes usarlo como tu editor de código.

```
vim /home/leonel/archivo.txt
```

11. Comando “pwd”

El comando pwd se usa para localizar la ruta del directorio de trabajo en el que te encuentras. Por ejemplo, si mi nombre de usuario es «miriam» y estoy en mi directorio Documentos, la ruta absoluta sería: /home/miriam/Documents.

```
pwd
```

12. Comando “mkdir”

Este comando permite a los usuarios crear directorios (carpetas). Posibilita la creación de varios directorios a la vez.

Parametros de mkdir

-p -> permite la creación de directorios dentro de directorios creados.

-v -> Muestra el estado de creación de directorios.

mkdir nombreDirectorio -> crea un directorio simple

o

mkdir -pv dir1/dir2/dir3 -> crea directorio dentro de otros directorio

13. Comando “find”

Los nuevos usuarios de Linux suelen decir que les confunde la ubicación de sus archivos en un servidor. Esto puede deberse a que la mayoría de la gente está acostumbrada a operar con Windows o MacOS, que tienen un diseño de directorios más claro y fácil de usar.

Aunque hay algo de verdad en esto, Linux da a los usuarios más opciones para buscar archivos usando ciertos comandos

Sintaxis básica

find <startingdirectory> <options> <search term>

Comienza con la palabra clave find, que alerta a Linux de que lo que sigue se refiere a la búsqueda de un archivo. El argumento <startingdirectory> es el punto de origen de donde deseas iniciar la búsqueda. Puede ser reemplazado con varios argumentos, incluyendo:

/ (slash) – busca en todo el sistema.

. (punto) – busca en la carpeta en la que estás trabajando actualmente (directorio actual).

~ (tilde) – para buscar desde tu directorio home.

Buscar un directorio o un fichero

find . -name my-file

Para buscar un directorio o fichero ignorando mayúsculas o minúsculas

find . -iname my-file

Para buscar directories o ficheros ignorando al objetivo

find . -not -name my-file

Para buscar varios archivos con un formato común .txt

find . -name “*.txt”

14. Comando “rm”

rm elimina cada archivo especificado en la línea de comando y directorios. Ten mucho cuidado al usarlo porque no se puede deshacer y es muy difícil recuperar archivos eliminados de esta manera.

Parámetros de “rm”

-r	->	Eliminar directorios de forma recursiva
-d	->	Eliminar directorios vacíos
-v	->	Muestra los elementos que se están eliminando
-i	->	nos pregunta si estamos seguros de eliminar un elemento

Para eliminar un archivo normal, debe escribir la siguiente sintaxis:

```
rm -driv file.txt
```

15. Comando “diff”

El comando diff nos ayuda a comparar dos archivos línea por línea y mostrar la diferencia entre ellos. Esta utilidad de línea de comandos lista los cambios que debe aplicar para que los archivos sean iguales.

```
diff documento.txt documento2.txt
```

16. Comando “chown”

chown permite cambiar el propietario del archivo o directorio. También se puede usar para cambiar la propiedad del grupo para el archivo o directorio.

Para cambiar propietario y grupo de un fichero

```
chown {nombreUsuario}:{nombreGrupo} {directorio/fichero}
```

Para cambiar solo propietario

```
chown {nombreUsuario} {directorio/fichero}
```

Para cambiar solo grupo

```
chown :{nombreGrupo} {directorio/fichero}
```

17. Comando “wget”

Este comando permite descargar archivos de Internet. Admite protocolos de red populares como FTP, HTTP y HTTPS. También puede manejar proxies HTTP.

```
wget url
```

18. Comando “top”

El comando top muestra una vista de los procesos en ejecución en Linux en tiempo real y muestra las tareas administradas por el kernel . También, muestra un resumen de información del sistema para ver la utilización de recursos, memoria y uso de CPU.

```
top
```

19. Comando "grep"

El comando grep «global regular expresión print» filtra el contenido de un archivo para facilitar nuestra búsqueda.

La forma más común de usarlo es:

```
ls -lah | grep ficheroBuscar
```

20. Comando "kill"

Las tareas en Linux se denominan procesos. Cada proceso tiene una ID de proceso única. Para finalizar un proceso en Linux podemos usar el comando kill.

El comando kill te permite terminar un proceso utilizando un ID de proceso específico, también conocido como PID. Para mostrar un pid en Linux, puedes ejecutar uno de los siguientes comandos. **ps, top**

sudo ps aux

Listara todos los procesos, sus PID, del usuario y el estado del proceso

```
root@prueba:/home/leonei# ps
  PID TTY          TIME CMD
 49777 pts/0    00:00:00 sudo
 49780 pts/0    00:00:00 su
 49781 pts/0    00:00:00 bash
 50534 pts/0    00:00:00 ps
```

sudo top

el comando TOP te permite ver las tareas/procesos en tiempo real y con más detalle, proporciona un resumen del sistema para verifacas si algo se destaca que pueda estar causando problemas en el servidor.

```
top - 19:44:41 up 21:24, 3 users, load average: 0.02, 0.01, 0.00
Tasks: 206 total, 1 running, 205 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 1.0 sy, 0.0 ni, 99.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 946.0 total, 76.4 free, 237.7 used, 631.9 buff/cache
MiB Swap: 3908.0 total, 3897.4 free, 10.6 used, 528.1 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 50547 root        20   0   8060  3744 3204 R  0.7   0.4   0:00.05 top
   8 root        0 -20    0    0    0 I  0.3   0.0   0:19.70 kworker/0:1H-events_highp+
  11 root        20   0    0    0    0 I  0.3   0.0   0:24.29 rcu_sched
  679 root        20   0 235748 6484 5104 S  0.3   0.7   2:05.49 vmtoolsd
 49180 root        20   0    0    0    0 I  0.3   0.0   0:05.92 kworker/0:1-events
   1 root        20   0 101832 10140 7320 S  0.0   1.0   0:04.89 systemd
   2 root        20   0    0    0    0 S  0.0   0.0   0:00.00 kthreadd
   3 root        0 -20    0    0    0 I  0.0   0.0   0:00.00 rcu_gp
   4 root        0 -20    0    0    0 I  0.0   0.0   0:00.00 rcu_par_gp
   6 root        0 -20    0    0    0 I  0.0   0.0   0:00.00 kworker/0:0H-events_highp+
```

Para finalizar un proceso específico con un PID identificado usar el siguiente comando

```
sudo kill 63772
```

A veces el comando anterior podría no funcionar, en ese caso quizás tengamos que forzar la cancelación del proceso, para cancelar un proceso de forma forzada debemos enviar una señal al comando kill

Hay múltiples señales disponibles en Linux que pueden usarse para interrumpir, finalizar o pausar procesos, para listar las señales que se pueden enviar a un proceso junto con el comando kill se debe usar el siguiente comando:

```
sudo kill -l
```

```
root@prueba:/home/leonel# kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS     8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

de esa forma podemos ver todas las señales

Para cancelar un proceso con el comando kill usando seniales se lo hace de la siguiente forma

```
sudo kill SIGKILL 63772
```

o de forma más reducida podemos usar:

```
sudo kill -9 63772
```

21. Comando “ping”

Ping es un comando o una herramienta de diagnóstico que permite hacer una verificación del estado de una determinada conexión o host local.

Si estás familiarizado con el lenguaje de redes es muy probable que te hayas topado con las letras Ping, un acrónimo para Packet Internet Groper, lo que literalmente en castellano sería “buscador de paquetes en redes”.

```
ping 8.8.8.8
ping google.com
```

22. Comando “shutdown”

El comando shutdown en nuestros sistemas GNU/Linux nos permitirá detener, apagar o reiniciar el sistema de una manera segura e incluso hacerlo a una hora determinada.

```
sudo shutdown
```

Esto hará que el sistema se apague de manera predeterminada en 1 minuto y mostrará una notificación de la hora a la que se realizará el apagado del sistema

```
sudo shutdown --halt +30
```

Esto hará que el sistema se apague de forma segura dentro de 30 minutos

```
sudo shutdown -h now
```

Esto hará que el sistema se apague de forma inmediata respetando el proceso de apagado

```
sudo shutdown --reboot 22:00
```

Esto hará que el sistema se reinicie a las 22:00 hora del sistema

```
sudo shutdown -r now
```

reinicia el sistema de forma inmediata

23. Comando “df”

El comando df se usa para mostrar la información relativa al espacio total y disponible de un sistema de archivos. Así que se puede usar junto con la opción -h para tener un resultado más humano.

```
sudo df -h
```

```
root@prueba:/home/leonel# df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                     430M         0   430M   0% /dev
tmpfs                     95M       1.3M    94M   2% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 257G       7.7G   237G   4% /
```

24. Comando “du”

Muestra el espacio usado por un directorio

```
sudo du -sh /var
```

mostrará el espacio usado por el directorio “/var” con una unidad legible para humanos

```
root@prueba:/home/leonel# du -sh /var
1.2G    /var
```

25. Comando “lsof”

Muestra archivos abiertos por un proceso

```
sudo lsof -i :80
```

mostrara los procesos que están usando el puerto 80

```
root@prueba:/home/leonel# lsof -i :80
COMMAND  PID    USER  FD   TYPE DEVICE SIZE/OFF NODE NAME
apache2  53001   root   4u    IPv6 363122 0t0    TCP *:http (LISTEN)
apache2  53002 www-data 4u    IPv6 363122 0t0    TCP *:http (LISTEN)
apache2  53003 www-data 4u    IPv6 363122 0t0    TCP *:http (LISTEN)
```