

Actividad | #2 | Primeros

Comandos de Shell

Sistemas Operativos I

Ingeniería en Desarrollo de
Software

TUTOR: Aarón Iván Salazar Macías

ALUMNO: Jesel Valle Rivera

FECHA: 20 de diciembre de 2024

Índice

Índice.....	2
Introducción.....	3
Descripción.....	4
¿Qué es un sistema operativo?.....	4
¿Qué es Linux?.....	4
¿Qué es una terminal?.....	5
¿Qué son los comandos?.....	5
Justificación.....	5
Desarrollo.....	7
Ejecución de comandos.....	7
Conclusión.....	15
Referencias.....	16

Introducción

En la era digital, los sistemas operativos basados en Linux han ganado una gran popularidad debido a su flexibilidad, seguridad y capacidad de personalización. “Algunas de sus características principales son: variedad de distribuciones, multiusuario y multitarea, estabilidad y seguridad, personalización y mayor control de los procesos”. (Felipe)

La línea de comandos, o terminal, es una interfaz fundamental para interactuar con estos sistemas. A través de ella, podemos realizar una amplia variedad de tareas, desde la gestión de archivos y directorios, hasta la configuración del sistema.

En esta actividad pondremos en práctica lo aprendido referente al Sistema Operativo Linux , su terminal y algunos comandos básicos útiles que se estarán ejecutando para visualizar la acción y la importancia de cada uno de ellos.

La lista de comandos en Linux, es realmente grande, pero nos enfocaremos en algunos de los más conocidos, y que nos ayudarán a realizar tareas cotidianas en poco tiempo.

Vamos a adentrarnos en este mundo de Linux y su terminal.

Descripción

¿Qué es un sistema operativo?

Se conoce por sistema operativo al software principal de una computadora o dispositivo que administra, gestiona, coordina y dirige los recursos del hardware (procesador, memoria, almacenamiento, etc) permitiendo así, que las aplicaciones y servicios se ejecuten. En otras palabras, es el puente entre el usuario y la máquina.

Algunos de los sistemas operativos más conocidos y utilizados son: Windows, macOS, Android, iOS y Linux. Esta vez nos enfocaremos en el último.

¿Qué es Linux?

Como hemos visto en la unidad, Linux es un conjunto de sistemas operativos de código abierto.

“Los principios de desarrollo de software de código abierto son:

- Transparencia
- Colaboración
- Lanzamiento temprano y frecuente
- Meritocracia inclusiva
- Comunidad

El modelo de desarrollo de código abierto también implica que las mejoras provienen de muchos contribuyentes corporativos e individuales diferentes, por lo que el rumbo del producto viene determinado en gran medida por la comunidad de usuarios y no por un único equipo de desarrollo de un solo proveedor”. (Oracle México)

¿Qué es una terminal?

La terminal es una interfaz de texto donde podemos comunicarnos directamente con el sistema operativo. Para ello, hacemos uso de comandos que ejecutan tareas específicas.

En Linux, la terminal es comúnmente utilizada para configuraciones y tareas avanzadas.

¿Qué son los comandos?

Los comandos son instrucciones que escribimos a través de la terminal para que el sistema operativo realice una acción específica. Cada comando tiene una función en particular y existe la posibilidad de combinarse con opciones o argumentos para tareas más complejas.

Justificación

Una terminal permite al usuario interactuar con el sistema operativo, lo cual facilita la automatización de tareas, administración de sistemas, desarrollo de software, entre otros.

“Los comandos de Linux permiten controlar el sistema desde la interfaz de línea de comandos (CLI) en lugar de utilizar un ratón o trackpad. Son instrucciones de texto introducidas en la terminal para indicarle al sistema exactamente qué hacer.

Los comandos que se ingresen en el terminal de Linux distinguen entre mayúsculas y minúsculas y siguen una sintaxis como “**comando -opciones argumentos**”. Se pueden combinar para tareas complejas mediante pipelines y redirecciones.

Algunas cosas clave para saber sobre los comandos de Linux:

- Son sensibles a mayúsculas y minúsculas; por ejemplo, “ls” y “LS” significan cosas diferentes.
- Siguen una sintaxis específica como “**comando -opciones argumentos**”.

- Se pueden combinar para realizar operaciones complejas mediante pipelines y redirecciones.
 - Proporcionan un control detallado sobre el sistema, algo difícil de lograr con interfaces gráficas.
 - Permiten automatizar tareas mediante scripts de shell y procesamiento por lotes.
 - Se pueden utilizar para acceder a recursos del sistema como el sistema de archivos, red, memoria y CPU.
 - Forman la base de la interacción con servidores y sistemas operativos Linux”.
- (DreamHost)

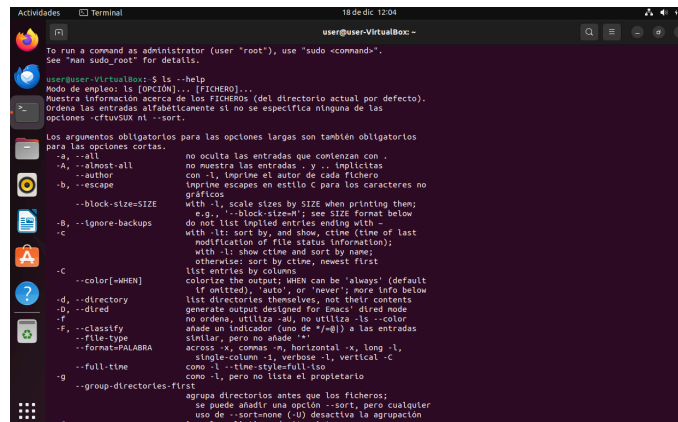
Desarrollo

Ejecución de comandos

1. Comando **ls**: nos ayuda a listar los archivos y directorios que hay en nuestro sistema. Su sintaxis es **ls [/directorio/carpeta/ruta]**; sin embargo, también se puede ejecutar sin una ruta y el comando mostrará el contenido del directorio actual.

El comando **ls** tiene varios parámetros de acuerdo a la acción específica que se quiera ejecutar. Algunos de los más usados son:

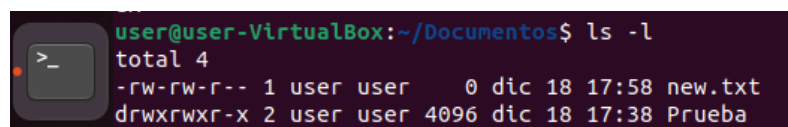
- ❖ **-help**: solicita al sistema que muestre un manual de instrucciones detallado sobre el comando **ls**. Al consultar el manual, puedes descubrir opciones que desconocías y adaptar el comando a tus necesidades particulares.



```
user@user-VirtualBox:~$ ls --help
Modo de empleo: ls [OPCIÓN]... [FICHERO]...
Muestra información acerca de los FICHEROS (del directorio actual por defecto).
ordena las entradas alfabéticamente si no se especifica ninguna de las
opciones -cfuvSUx nl --sort.

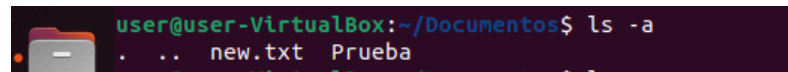
Los argumentos obligatorios para las opciones largas son también obligatorios
para las opciones cortas.
-a, --all                no oculta las entradas que comienzan con .
-A, --almost-all       no muestra las entradas . y .. implícitas
                        con -l, imprime el autor de cada fichero
-b, --escape            imprime escapes en estilo C para los caracteres no
                        gráficos
--block-size=SIZE       with -l, scale sizes by SIZE when printing them;
                        e.g., '--block-size=M'; see SIZE format below
                        do not list implied entries ending with
-B, --ignore-backups    with -l: sort by, and show, ctime (time of last
                        modification of file status information);
                        with -lt: show ctime and sort by name;
                        otherwise: sort by ctime, newest first
-C                      list entries by columns
--color[=WHEN]          colorize the output; WHEN can be 'always' (default
                        if omitted), 'auto', or 'never'; more info below
-d, --directory         list directories themselves, not their contents
-D, --dired              generate output designed for Emacs' dired mode
-f                      no ordena, utiliza -du, no utiliza -ls --color
-F, --classify           añade un indicador (uno de */@|) a las entradas
                        similar, pero no añade el
--file-type              across *, commas →, horizontal -x, long -l,
                        single-column -l, verbose -l, vertical -C
--format=PALABRA        como -l --time=styleFulliso
                        como -l, pero no lista el propietario
-g, --group-directories-first
                        agrupa directorios antes que los ficheros;
                        se puede añadir una opción --sort, pero cualquier
                        uso de --sortnone (-U) desactiva la agrupación
-G, --no-group           to a long listing, don't print group names
```

- ❖ **-l**: “muestra los resultados en formato largo, mostrando detalles adicionales como permisos, propiedad, tamaño y fecha de modificación para cada archivo y directorio”. (DreamHost)



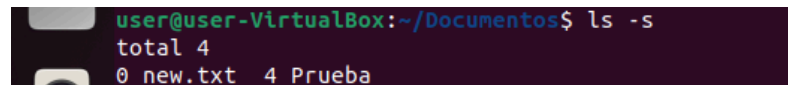
```
user@user-VirtualBox:~/Documentos$ ls -l
total 4
-rw-rw-r-- 1 user user  0 dic 18 17:58 new.txt
drwxrwxr-x 2 user user 4096 dic 18 17:38 Prueba
```

- ❖ -a: “Muestra archivos y directorios ocultos que comienzan con . además de los elementos no ocultos”. (DreamHost)



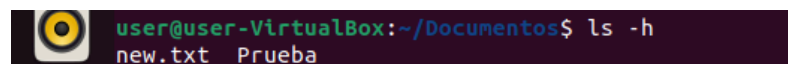
```
user@user-VirtualBox:~/Documentos$ ls -a
.  ..  new.txt  Prueba
```

- ❖ -s: se utiliza para listar los archivos y directorios en un directorio específico, mostrando el tamaño en bloques de cada archivo o directorio.



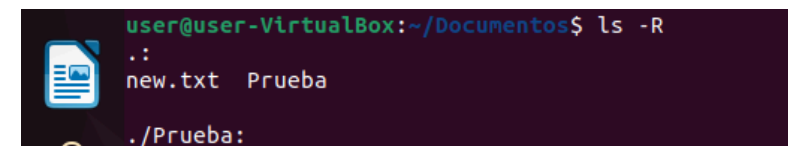
```
user@user-VirtualBox:~/Documentos$ ls -s
total 4
0 new.txt  4 Prueba
```

- ❖ -h: se utiliza para listar los archivos y directorios en un directorio específico, mostrando el tamaño de cada archivo o directorio en formato humano-legible. Esto significa que los tamaños se muestran en unidades como kilobytes (KB), megabytes (MB), gigabytes (GB), etc., en lugar de bytes.



```
user@user-VirtualBox:~/Documentos$ ls -h
new.txt  Prueba
```

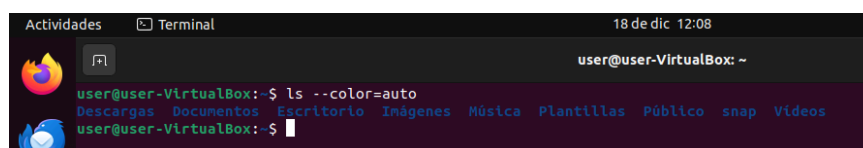
- ❖ -R: explorará cada subdirectorio y listará sus contenidos, y así sucesivamente hasta llegar a los subdirectorios más profundos. “Lista todos los archivos de los subdirectorios”. (A.)



```
user@user-VirtualBox:~/Documentos$ ls -R
.:
new.txt  Prueba

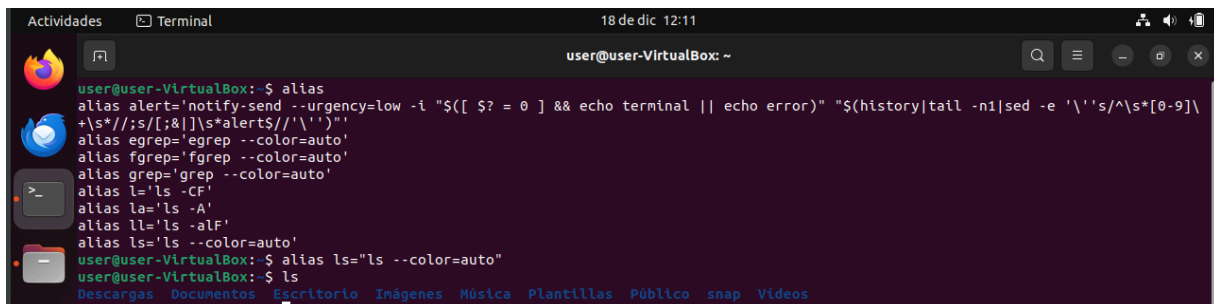
./Prueba:
```

- ❖ -ls --color=auto: se utiliza para listar archivos y directorios en la terminal, pero con la particularidad de colorear la salida para mejorar la legibilidad.



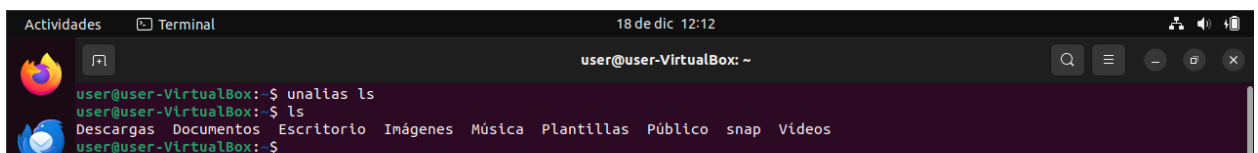
```
Actividades  Terminal  18 de dic 12:08
user@user-VirtualBox: ~
user@user-VirtualBox:~$ ls --color=auto
Descargas Documentos Escritorio Imágenes Música Plantillas Público snap Videos
user@user-VirtualBox:~$
```


2. Comando **alias**: crea accesos directos para comandos largos o complejos. (DreamHost). Es como crear un apodo para un amigo: en lugar de decir su nombre completo, puedes usar el apodo y todos sabrán a quién te refieres. Su sintaxis básica es **alias nombre_alias='comando_original'**.



```
user@user-VirtualBox: ~  
$ alias  
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error" "${history|tail -n1|sed -e '\''s/^s*[0-9]\n+/s*//;s/[;]&|s*alert$//'\`}'"  
alias egrep='egrep --color=auto'  
alias fgrep='fgrep --color=auto'  
alias grep='grep --color=auto'  
alias l='ls -CF'  
alias la='ls -A'  
alias ll='ls -aF'  
alias ls='ls --color=auto'  
user@user-VirtualBox:~$ alias ls="ls --color=auto"  
user@user-VirtualBox:~$ ls  
Descargas Documentos Escritorio Imágenes Música Plantillas Público snap Videos
```

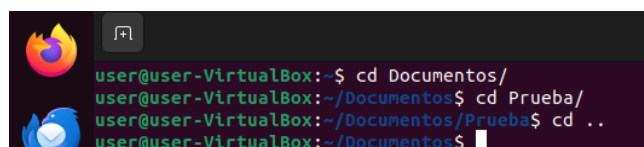
3. Comando **unalias**: contrariamente al anterior, se utiliza para eliminar los alias que se hayan creado previamente. Su sintaxis básica es: **unalias nombre_alias**. Ojo: Si quieres eliminar todos los alias de una vez, puedes usar la opción -a: **unalias -a**



```
user@user-VirtualBox:~$ unalias ls  
user@user-VirtualBox:~$ ls  
Descargas Documentos Escritorio Imágenes Música Plantillas Público snap Videos  
user@user-VirtualBox:~$
```

4. Comando **cd**: se utiliza para navegar entre directorios. Te permite moverte desde el directorio de trabajo actual a una nueva ubicación en el sistema de archivos. Cuando ejecutas el comando **cd** por sí mismo, te llevará al directorio de inicio. También puedes pasar una ruta específica para cambiarte a ella. Su sintaxis básica es **cd [directorio]**.

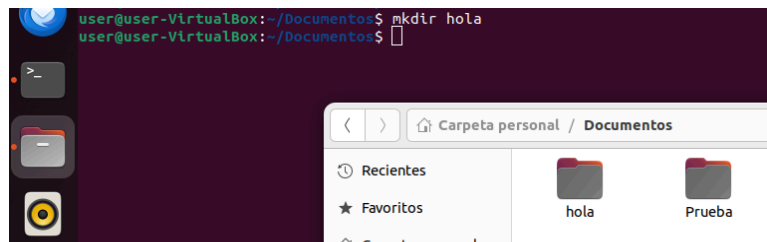
- ❖ **cd ..** : te mueve un nivel atrás hacia el directorio principal
- ❖ **cd -** : cambia al directorio previo o anterior



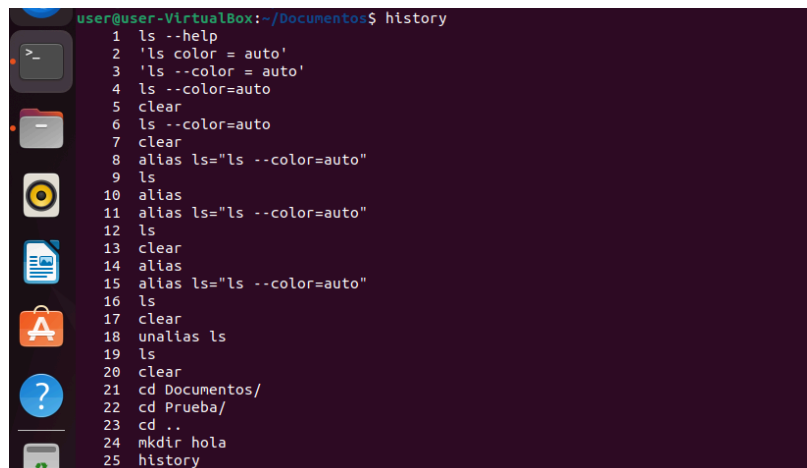
```
user@user-VirtualBox:~$ cd Documentos/  
user@user-VirtualBox:~/Documentos$ cd Prueba/  
user@user-VirtualBox:~/Documentos/Prueba$ cd ..  
user@user-VirtualBox:~/Documentos$
```

5. Comando **mkdir**: crea uno o varios directorios. Su sintaxis básica es **mkdir [opción] <directorio>**.

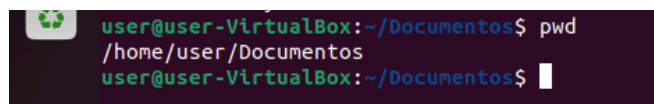
- ❖ **-p** : crea un directorio entre dos carpetas existentes. Por ejemplo, `mkdir -p Música/2024/Canciones` crea un nuevo directorio 2024.
- ❖ **-m**: establece los permisos de la carpeta. Por ejemplo, introduce `mkdir -m777 directory` para crear un directorio con permisos de lectura, escritura y ejecución para todos los usuarios.
- ❖ **-v**: imprime un mensaje por cada directorio creado.



6. Comando **history**: enlista o “muestra los comandos utilizados previamente en la sesión actual del terminal”. (DreamHost)

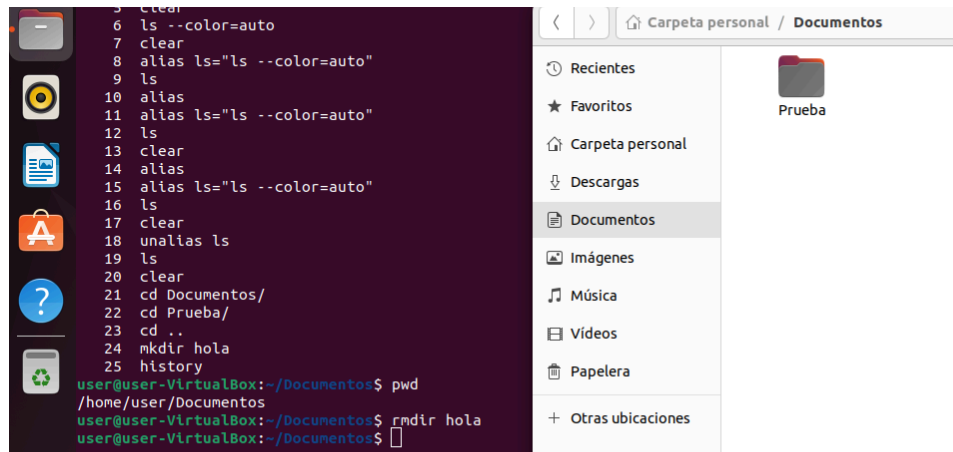


7. Comando **pwd**: “imprime la ruta del directorio de trabajo actual”. (A.)



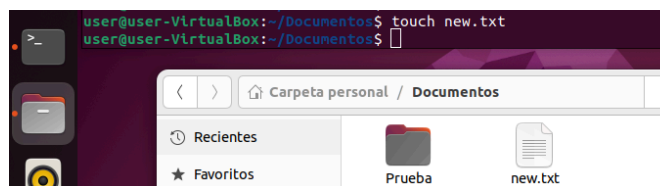
8. Comando **rmdir**: este comando elimina directorios vacíos. Su sintaxis básica es **rmdir [opciones] <directorío>**. Si la carpeta contiene un subdirectorio, el comando devolverá un error. Para eliminar directorios con contenido, se usa el comando **rm**.

- ❖ **-v**: muestra un resultado detallado al eliminar directorios.
- ❖ **-p**: fuerza la eliminación de un directorio no vacío.



9. Comando **touch**: permite crear un archivo vacío en una ruta de directorio específica. Esto es útil cuando necesitas un archivo vacío para llenarlo con datos más tarde. Su sintaxis básica es **touch [opciones] /nombredearchivo**.

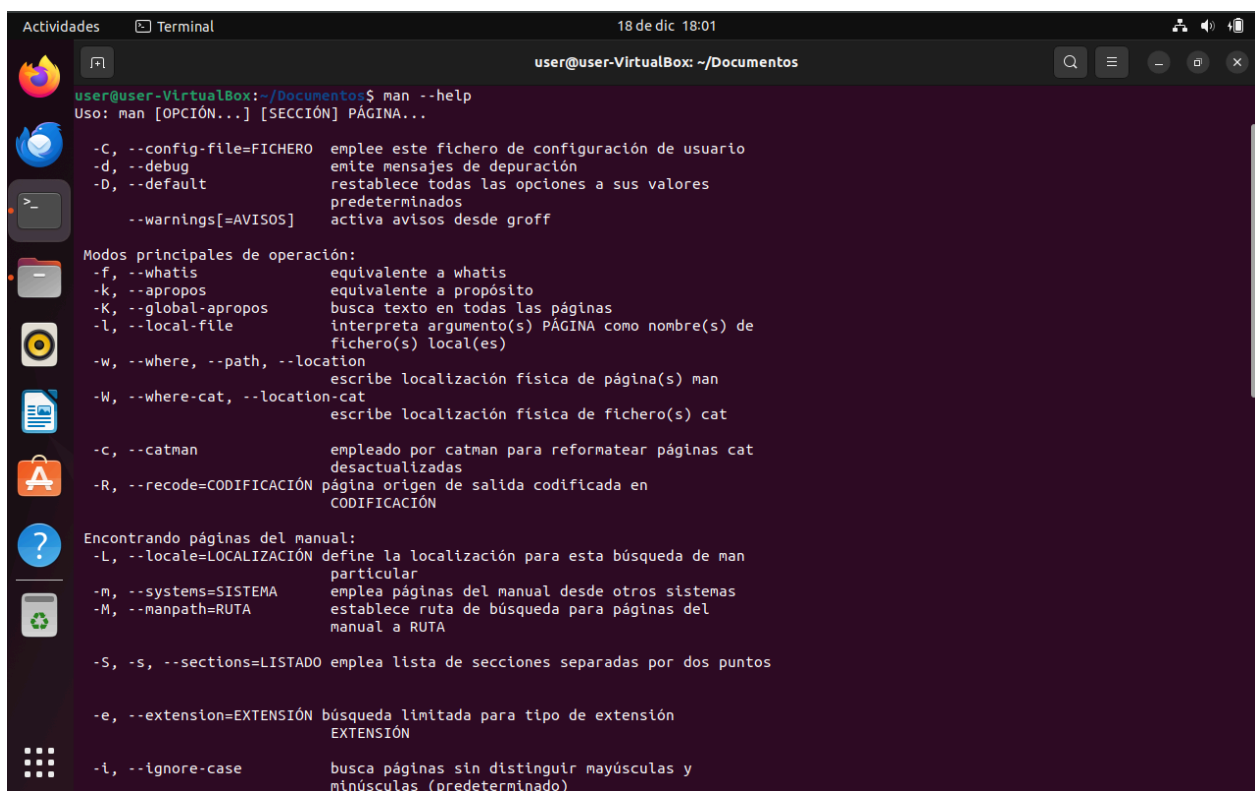
- ❖ **-c**: con esta opción se crea el archivo, solo si no existe. Evita sobrescribir archivos existentes accidentalmente.
- ❖ **-m**: “en lugar de crear un nuevo archivo, actualiza la marca de tiempo en un archivo existente. Esto se puede usar para cambiar la hora de modificación”. (DreamHost)



10. Comando **man --help**: esta es una combinación del comando man y el argumento u opción help. El comando man “proporciona información sobre los comandos u otros programas del entorno Linux (manual pages)”. (Facultad de Ingeniería). Su sintaxis básica es **man [opción] [número_sección] nombre_comando**.

Por otro lado, la opción help es sirve para solicitar ayuda o información sobre cómo usar el programa. Así que en resumen, esta combinación muestra cómo utilizar el comando man para buscar información sobre otros comandos.

- ❖ **ls**: lista el contenido de un directorio, si lo usamos sin argumentos o ruta, mostrará el contenido del directorio actual.
- ❖ **man**: sirve para usar el comando man de forma más precisa



```
user@user-VirtualBox:~/Documentos$ man --help
Uso: man [OPCIÓN...] [SECCIÓN] PÁGINA...

-C, --config-file=FICHERO  emplee este fichero de configuración de usuario
-d, --debug                emite mensajes de depuración
-D, --default              restablece todas las opciones a sus valores
                           predeterminados
--warnings[=AVISOS]       activa avisos desde groff

Modos principales de operación:
-f, --whatis               equivalente a whatis
-k, --apropos              equivalente a propósito
-K, --global-apropos       busca texto en todas las páginas
-l, --local-file           interpreta argumento(s) PÁGINA como nombre(s) de
                           fichero(s) local(es)
-w, --where, --path, --location  escribe localización física de página(s) man
-W, --where-cat, --location-cat  escribe localización física de fichero(s) cat

-c, --catman               empleado por catman para reformatear páginas cat
                           desactualizadas
-R, --recode=CODIFICACIÓN  página origen de salida codificada en
                           CODIFICACIÓN

Encontrando páginas del manual:
-L, --locale=LOCALIZACIÓN  define la localización para esta búsqueda de man
                           particular
-m, --systems=SISTEMA      emplea páginas del manual desde otros sistemas
-M, --manpath=RUTA         establece ruta de búsqueda para páginas del
                           manual a RUTA

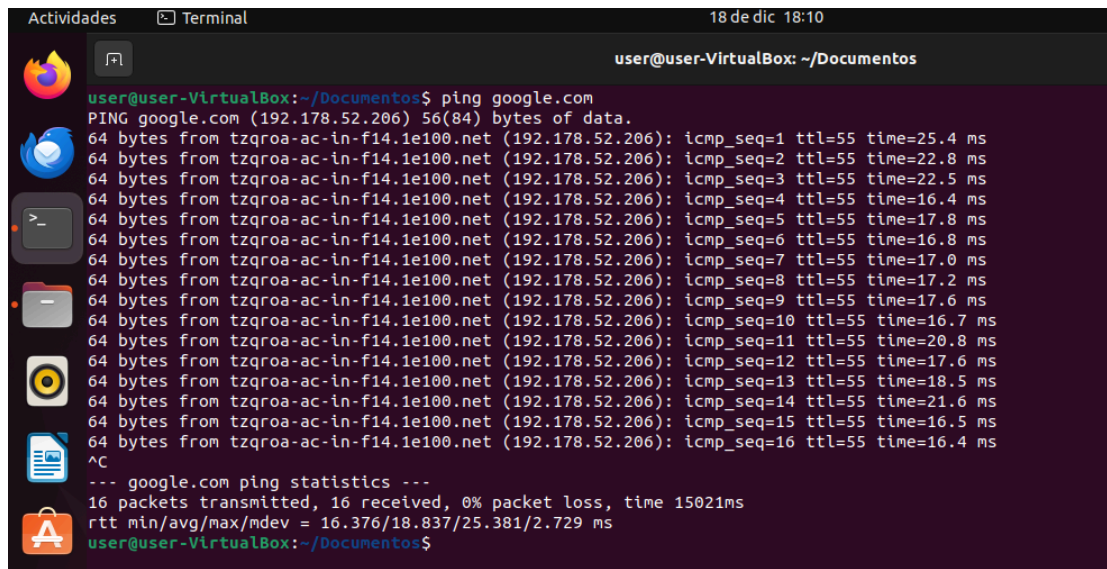
-S, -s, --sections=LISTADO emplea lista de secciones separadas por dos puntos

-e, --extension=EXTENSIÓN  búsqueda limitada para tipo de extensión
                           EXTENSIÓN

-i, --ignore-case          busca páginas sin distinguir mayúsculas y
                           minúsculas (predeterminado)
```

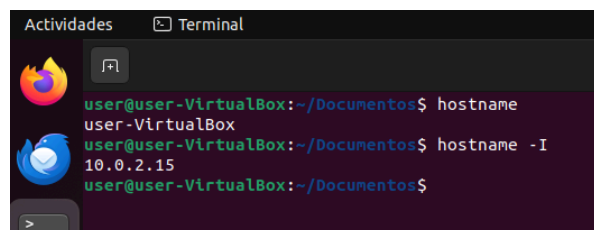
11. Comando **ping**: permite comprobar si se puede acceder a una red o a un servidor.

Esto lo hace enviando paquetes de solicitud de eco ICMP y escuchando las respuestas de eco. Su sintaxis básica es **ping [opción] [hostname_o_dirección_IP]**.



```
user@user-VirtualBox: ~/Documentos
user@user-VirtualBox:~/Documentos$ ping google.com
PING google.com (192.178.52.206) 56(84) bytes of data:
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=1 ttl=55 time=25.4 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=2 ttl=55 time=22.8 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=3 ttl=55 time=22.5 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=4 ttl=55 time=16.4 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=5 ttl=55 time=17.8 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=6 ttl=55 time=16.8 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=7 ttl=55 time=17.0 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=8 ttl=55 time=17.2 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=9 ttl=55 time=17.6 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=10 ttl=55 time=16.7 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=11 ttl=55 time=20.8 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=12 ttl=55 time=17.6 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=13 ttl=55 time=18.5 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=14 ttl=55 time=21.6 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=15 ttl=55 time=16.5 ms
64 bytes from tzqroa-ac-in-f14.1e100.net (192.178.52.206): icmp_seq=16 ttl=55 time=16.4 ms
^C
--- google.com ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15021ms
rtt min/avg/max/mdev = 16.376/18.837/25.381/2.729 ms
user@user-VirtualBox:~/Documentos$
```

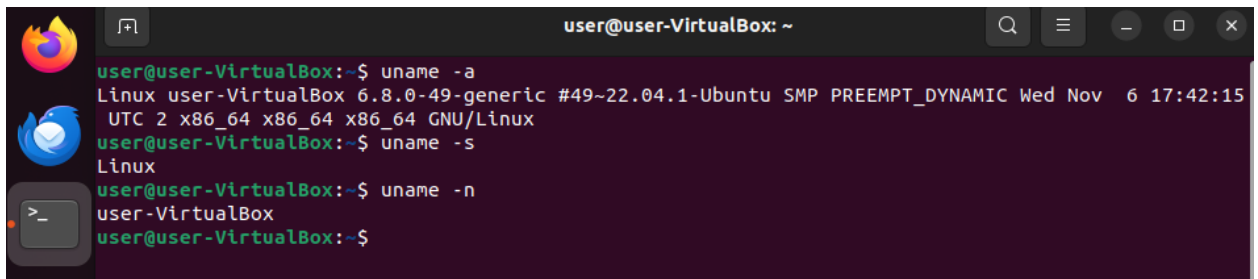
12. Comando **hostname** y **hostname -I**: el comando **hostname** imprime o configura el identificador del nombre de host para su sistema Linux en la red. Sin argumentos, muestra el nombre de host actual. La opción **-I** se utiliza para mostrar las direcciones IP de las interfaces de red activas en un sistema.



```
user@user-VirtualBox:~/Documentos$ hostname
user-VirtualBox
user@user-VirtualBox:~/Documentos$ hostname -I
10.0.2.15
user@user-VirtualBox:~/Documentos$
```

13. Comando **uname**: imprime información sobre nuestra máquina, incluyendo el hardware, el nombre del sistema y el núcleo Linux. Su sintaxis básica es: **uname [opción]**.

- ❖ **-a**: imprime toda la información del sistema.
- ❖ **-s**: muestra el nombre del núcleo.
- ❖ **-n**: muestra el nombre de host del nodo del sistema.

A terminal window titled 'user@user-VirtualBox: ~' with search, menu, and window control icons. It shows three commands and their outputs: 'uname -a' returns system details like kernel version and architecture; 'uname -s' returns 'Linux'; 'uname -n' returns 'user-VirtualBox'.

```
user@user-VirtualBox:~$ uname -a
Linux user-VirtualBox 6.8.0-49-generic #49~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Nov  6 17:42:15
UTC 2 x86_64 x86_64 x86_64 GNU/Linux
user@user-VirtualBox:~$ uname -s
Linux
user@user-VirtualBox:~$ uname -n
user-VirtualBox
user@user-VirtualBox:~$
```

Conclusión

A través de la ejecución de estos comandos básicos en la terminal de Linux, hemos podido explorar las capacidades básicas del sistema operativo desde una perspectiva práctica. Cada comando nos ha brindado una herramienta esencial para interactuar con el sistema de archivos, gestionar procesos, obtener información del sistema y realizar tareas administrativas.

En conclusión, la terminal de Linux es una poderosa herramienta que nos permite interactuar con el sistema operativo a un nivel muy profundo. Los comandos que hemos explorado en este trabajo son solo una pequeña muestra de las posibilidades que ofrece la línea de comandos. Al dominar estos comandos básicos, hemos sentado las bases para poder realizar tareas más complejas y automatizar procesos.

A modo de reflexión, es importante destacar que el uso de la terminal no solo es útil para administradores de sistemas, sino también para cualquier usuario de Linux que desee tener un mayor control sobre su sistema. Al comprender los fundamentos de la línea de comandos, podemos optimizar nuestro flujo de trabajo, resolver problemas de manera más eficiente y personalizar nuestro entorno de trabajo de manera práctica. Sigamos explorando la terminal y comandos más avanzados.

Referencias

A., Deyimar. “Los 40 comandos de Linux más utilizados en 2024.” *Hostinger*, 21 de noviembre de 2024, <https://www.hostinger.mx/tutoriales/linux-comandos>. Consultado el 19 de diciembre de 2024.

DreamHost. “Top 100 Comandos Linux (Que Debes Conocer).” *¿Qué son los comandos Linux?*, 18 de diciembre de 2023, <https://www.dreamhost.com/blog/es/comandos-linux-que-debes-conocer/>. Consultado el 19 de diciembre de 2024.

Facultad de Ingeniería. “Comandos Básicos de Shell (Terminal) | Facultad de Ingeniería.” *FING*, (s.f.), <https://www.fing.edu.uy/es/sysadmin/salas-estudiantiles-linux/comandos-b%C3%A1sicos-de-shell-terminal>. Consultado el 20 de diciembre de 2024.

Felipe. “Cuáles son las características de Linux.” *Hostingplus*, 19 de abril de 2021, <https://www.hostingplus.pe/blog/cuales-son-las-caracteristicas-de-linux/>. Consultado el 20 de diciembre de 2024.

Oracle México. “¿Qué es Linux?”, (s.f.), <https://www.oracle.com/mx/linux/what-is-linux/>. Consultado el 19 de diciembre de 2024.