

Modulo 1 –Tema 2. Terminal y línea de comandos de Linux.

Comandos para la navegación básica en la terminal
de Linux

Objetivos:

Proporcionar las habilidades y conocimientos necesarios para utilizar la terminal de Linux de forma eficiente en sus proyectos de análisis de datos.

Contenido

1.- Introducción a la terminal de Linux:

- Conceptos básicos: shell, comandos, argumentos, opciones
- Navegación por el sistema de archivos: cd, ls, pwd, mkdir, rmdir
- Edición de archivos: nano, vim
- Permisos de archivos y directorios: chmod, chown

2.- Comandos básicos de la terminal:

- grep: búsqueda de texto en archivos
- sort: ordenar líneas de un archivo
- uniq: eliminar líneas duplicadas
- wc: contar líneas, palabras y caracteres
- head/tail: mostrar las primeras/últimas líneas de un archivo
- tee: enviar la salida de un comando a un archivo y a la pantalla
- pipe: combinar la salida de dos comandos

Conceptos básicos. Shell de comandos

Linux dispone de una aplicación denominada intérprete de comandos o shell de comandos que permite al usuario interactuar con el sistema operativo mediante la ejecución de comandos o sentencias de texto. Hay diferentes tipos de intérpretes que se diferencian en la sintaxis de los comandos y en la forma de interaccionar con el usuario.

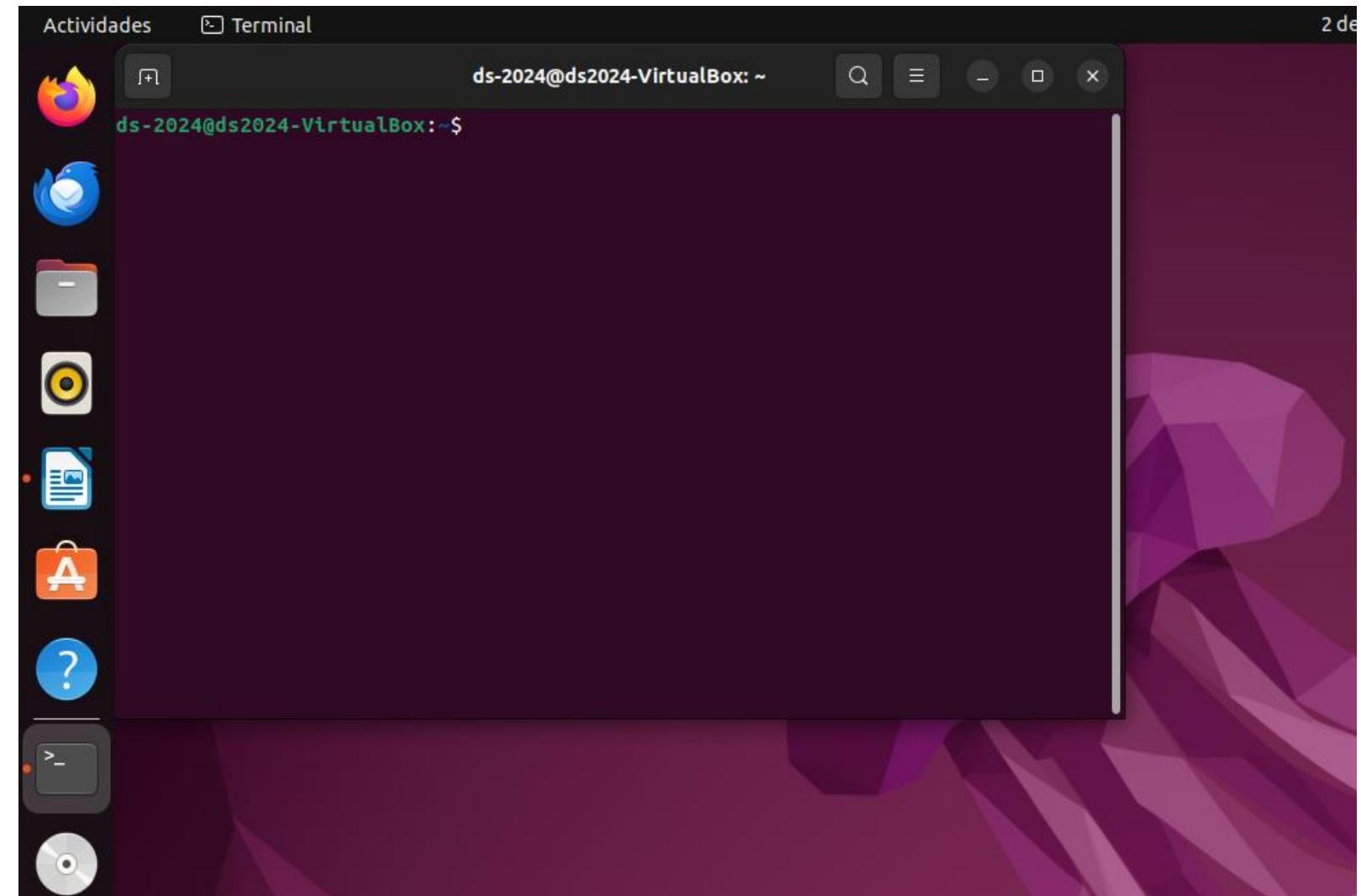
Conceptos básicos. Shell de comandos

Según el sistema Linux utilizado, habrá diferentes tipos de shell. La más utilizada es Bash (Bourne Again Shell), que no solo permite ejecutar comandos puntuales, sino que dispone de un sistema de sentencias donde se pueden componer scripts o automatismos de cierto grado de complejidad.

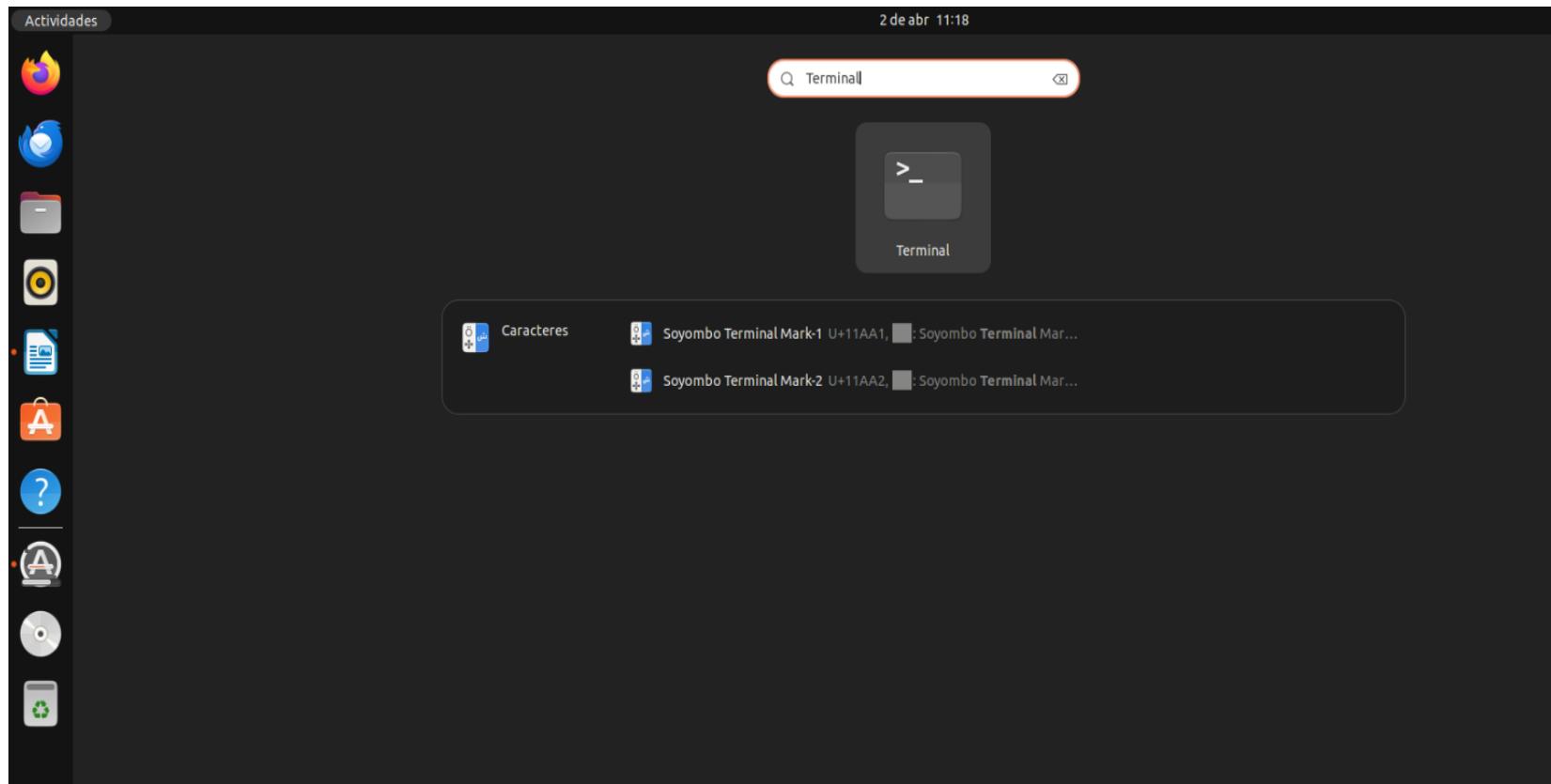
Conceptos básicos. Shell de comandos

Para acceder al intérprete de comandos, se pulsa Ctrl+Alt+T o bien se pulsa en un ícono que aparece en la parte inferior izquierda, dentro de la barra de herramientas inferior del escritorio de ubuntu

Para acceder al intérprete de comandos, se pulsa Ctrl+Alt+T, o bien se pulsa en un ícono que aparece en la parte inferior izquierda, dentro de la barra de herramientas inferior del escritorio de ubuntu

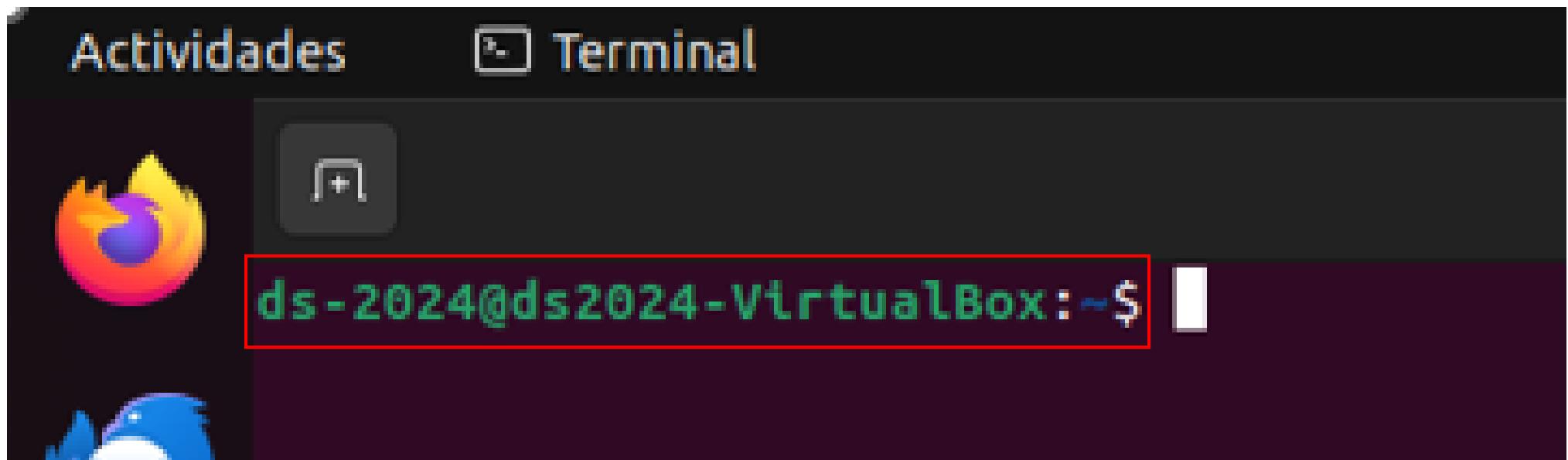


o bien se pulsa en un icono
que aparece en la parte
inferior izquierda,
de la barra de
herramientas inferior del
escritorio de Ubuntu



Algunas características generales son:

El terminal muestra en pantalla un indicador de línea de órdenes, denominado “prompt”, para que el usuario introduzca una. El indicador finaliza con un carácter \$ en el caso de usuarios normales y con # en el caso del superusuario.



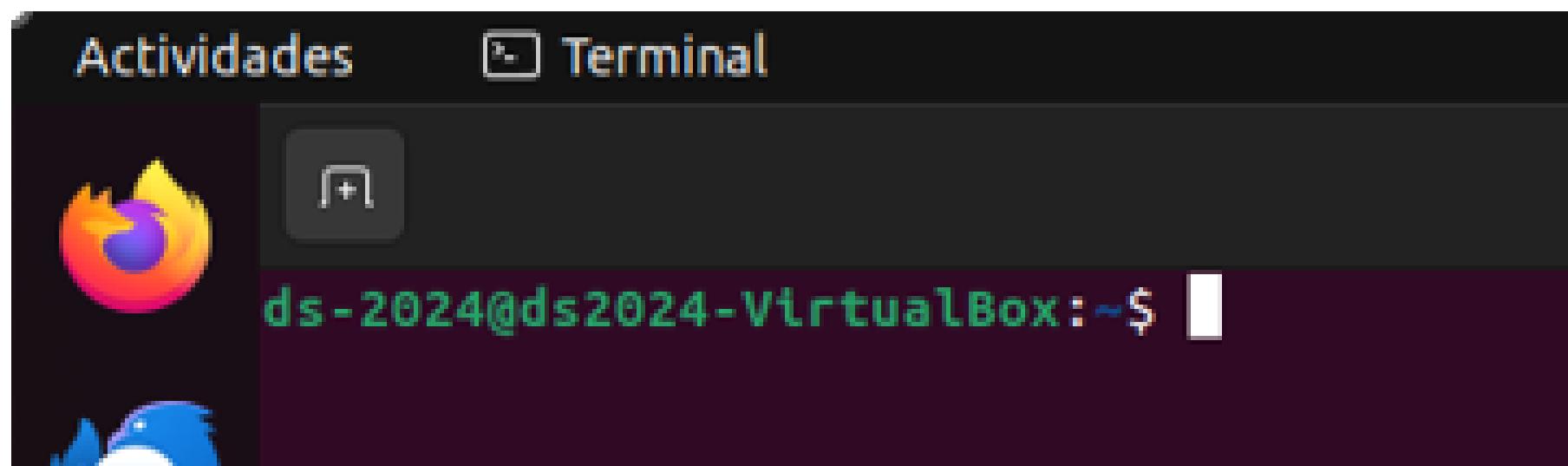
Al comienzo de la línea de órdenes aparece el usuario en la forma **usuario@máquina:directorio\$**, donde:

“usuario” es el nombre del usuario logado.

“máquina” es el nombre asignado a la máquina.

“directorio” es la ruta, dentro del sistema de ficheros, en la que se encuentra el usuario en ese momento.

El carácter especial “~” se utiliza para referirse al “HOME” del usuario, que normalmente se corresponde con la ruta “/home/<usuario>”. Es el espacio de ficheros dentro del sistema global asignado a ese usuario

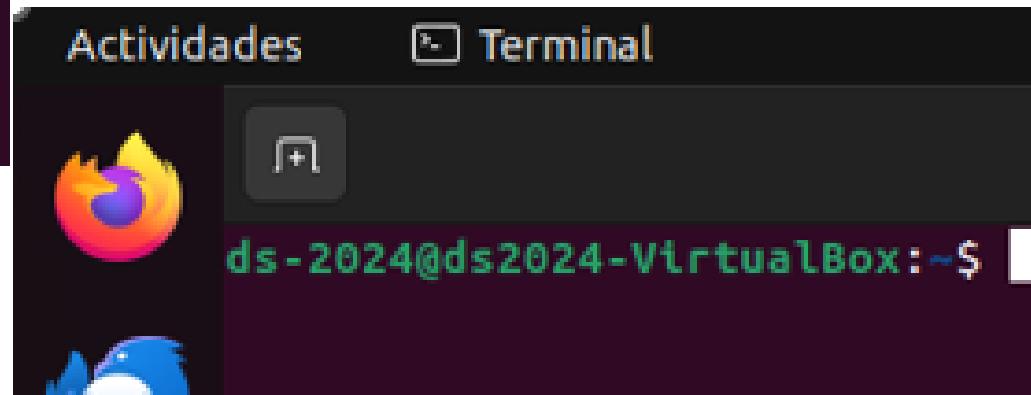
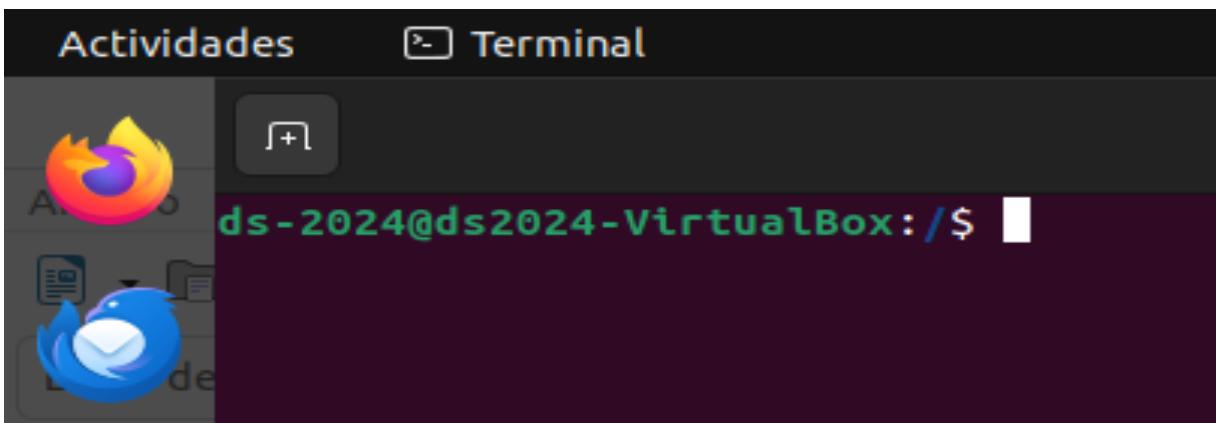


Algunas características generales son:

- Cuando se escribe un comando para que se ejecute, hay que pulsar la tecla Enter.
- Los comandos hay que teclearlos exactamente. En este sentido, las letras mayúsculas y minúsculas se consideran caracteres diferentes.
- Un amplio número de comandos de la shell está pensado para trabajar directamente con el sistema de ficheros (o sistema de archivos). Por tanto, es conveniente entender cómo se organiza este sistema y cómo la shell da información sobre él.

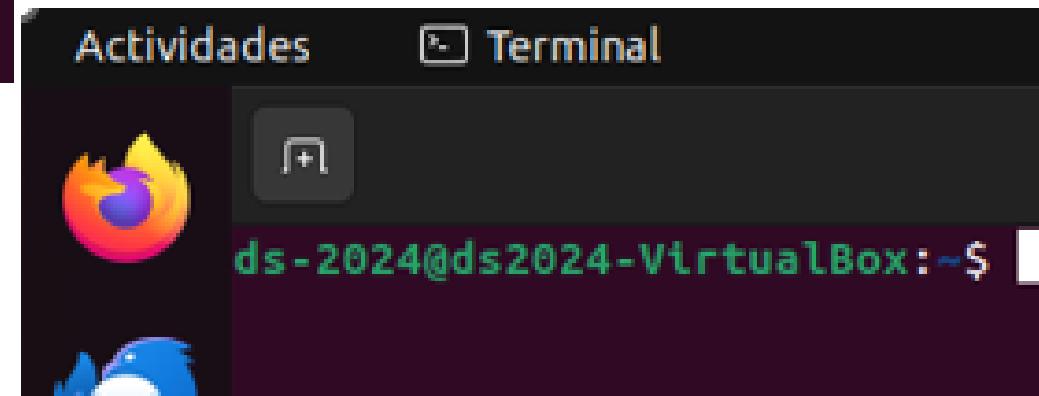
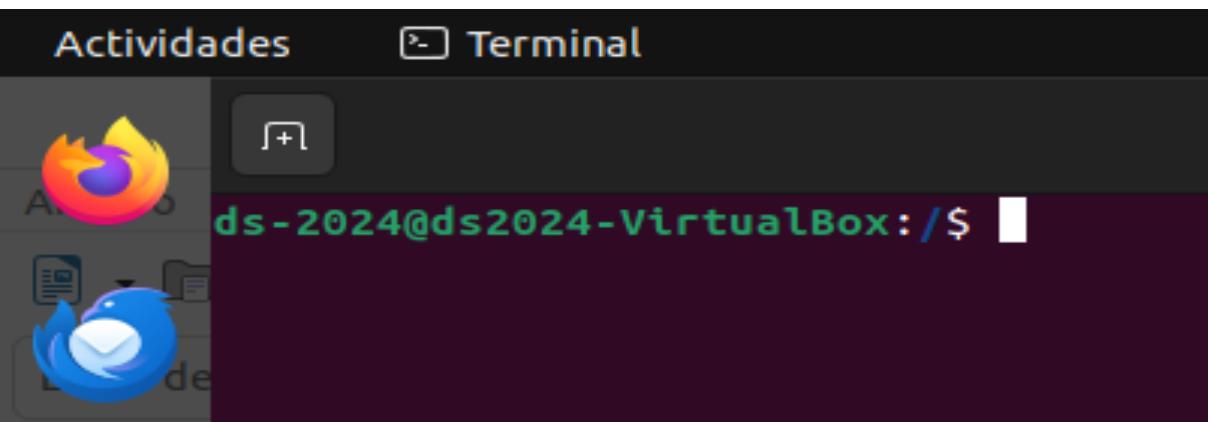
Algunas características generales son:

- El directorio raíz es “/” y es el punto de partida hacia cualquier otro fichero o directorio del sistema.
- La jerarquía de directorios se visualiza mediante rutas, donde se muestra el camino, de izquierda a derecha, que se ha seguido para llegar a un directorio. Por ejemplo, si hay un directorio “prueba1” en el HOME, el camino o “PATH” seguido será: “/home/usuario/prueba1”.



Algunas características generales son:

- El directorio raíz es “/” y es el punto de partida hacia cualquier otro fichero o directorio del sistema.
- La jerarquía de directorios se visualiza mediante rutas, donde se muestra el camino, de izquierda a derecha, que se ha seguido para llegar a un directorio. Por ejemplo, si hay un directorio “prueba1” en el HOME, el camino o “PATH” seguido será: “/home/usuario/prueba1”.



Estructura de los directorios

- * En el sistema de ficheros de UNIX (y similares, como GNU/Linux), existen varias sub-jerarquías de directorios que poseen múltiples y diferentes funciones de almacenamiento y organización en todo el sistema. Estos directorios pueden clasificarse en:
- * ° Estáticos: Contiene archivos que no cambian sin la intervención del administrador (root), sin embargo, pueden ser leídos por cualquier otro usuario. (/bin, /sbin, /opt, /boot, /usr/bin...)

Estructura

- * <^o Dinámicos: Contiene archivos que son cambiantes, y pueden leerse y escribirse (algunos sólo por su respectivo usuario y el root). Contienen configuraciones, documentos, etc. (/var/mail, /var/spool, /var/run, /var/lock, /home...)
- * <^o Compartidos: Contiene archivos que se pueden encontrar en un ordenador y utilizarse en otro, o incluso compartirse entre usuarios.

Estructura

- * <^o Restringidos: Contiene ficheros que no se pueden compartir, solo son modificables por el administrador. (/etc, /boot, /var/run, /var/lock...)

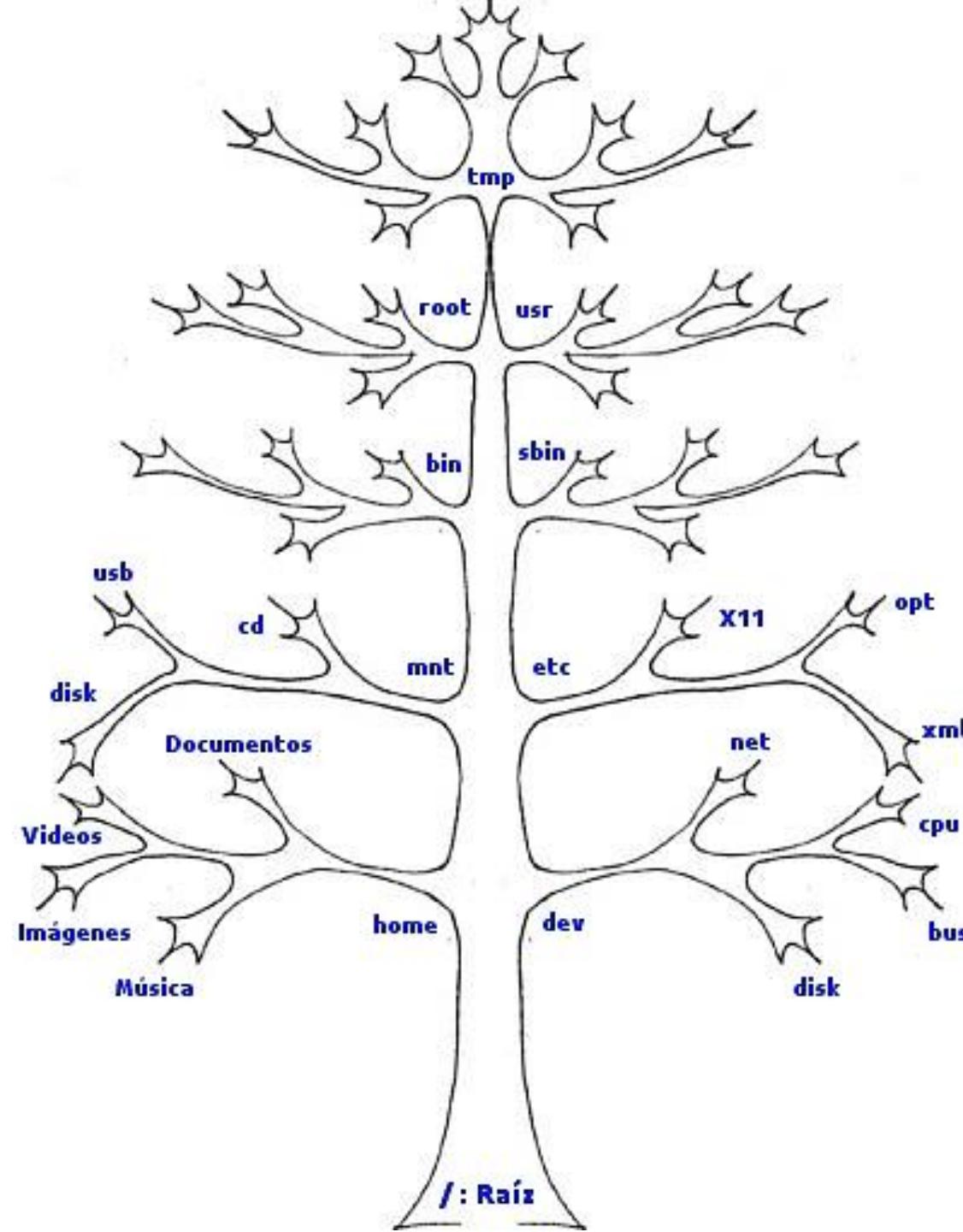
Estructura

- * root: es el nombre convencional de la cuenta de usuario que posee todos los derechos en todos los modos (mono o multi usuario). root es también llamado superusuario. Normalmente esta es la cuenta de administrador. El usuario root puede hacer muchas cosas que un usuario común no puede, tales como cambiar el dueño o permisos de archivos y enlazar a puertos de numeración pequeña.

Estructura

- * No es recomendable utilizar el usuario root para una simple sesión de uso habitual, ya que pone en riesgo el sistema al garantizar acceso privilegiado a cada programa en ejecución. Es preferible utilizar una cuenta de usuario normal y utilizar el comando su para acceder a los privilegios de root de ser necesario

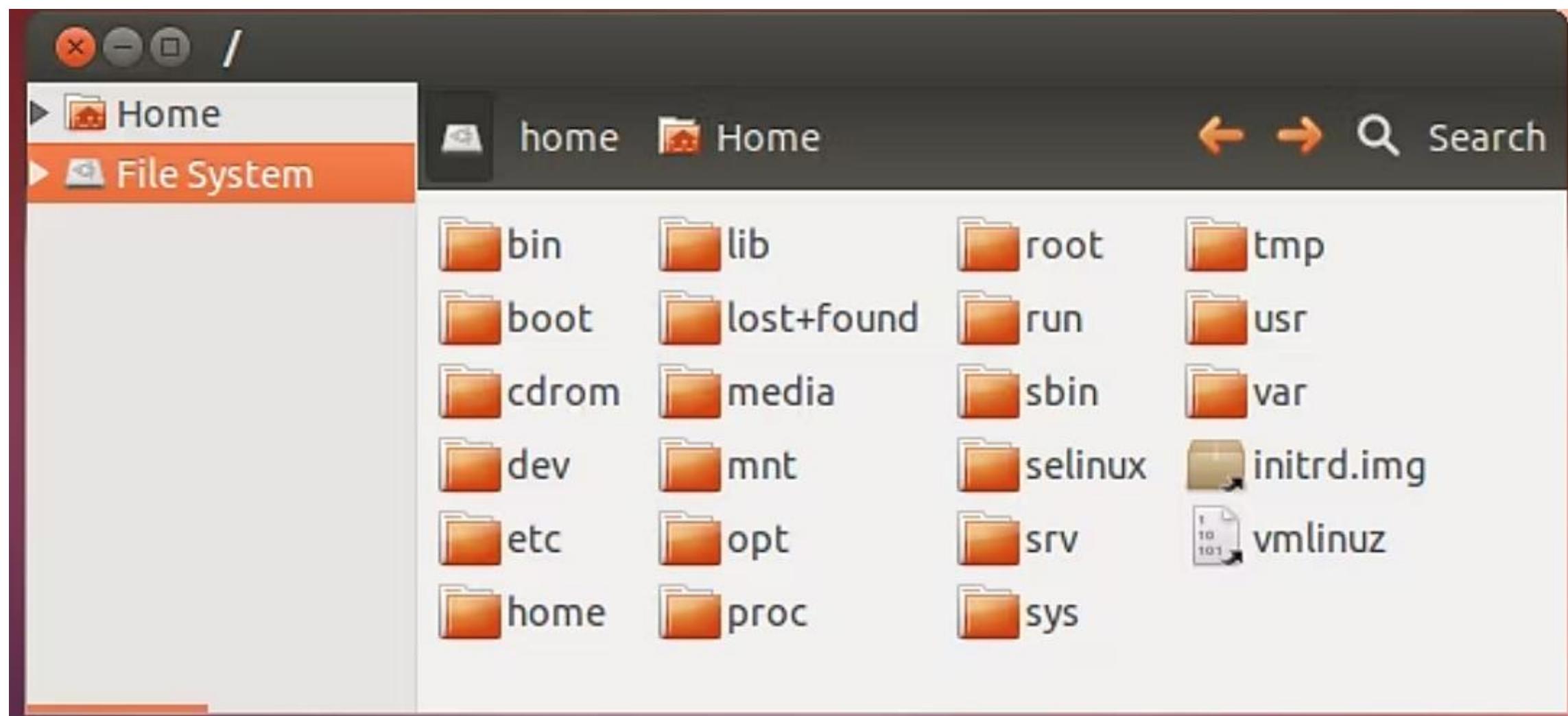
Donde la raíz del árbol (/) es la base de toda la estructura de directorios y las ramas (directorios y archivos) surgen o cuelgan de dicha base.



Estructura Standard

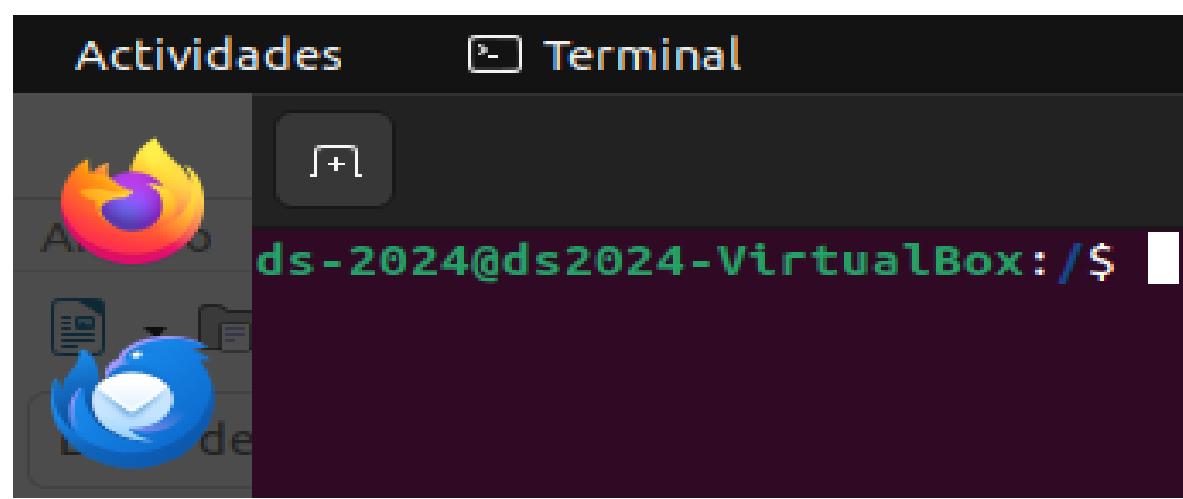


Sistema de archivos de Linux



Estructura Raíz ‘/’

/ (raíz): Parecido a el directorio raíz “C:\” de los sistemas operativos DOS y Windows. Es el nivel más alto dentro de la jerarquía de directorios, es el contenedor de todo el sistema (accesos al sistema de archivos, incluyendo los discos extraíbles [CD’s, DVD’s, pendrives, etc.]).



Estructura bin

/bin (binarios): Los binarios son los ejecutables de Linux (similar a los archivos.exe de Windows). Aquí tendremos los ejecutables de los programas propios del sistema operativo.

/bin/

Comandos binarios esenciales
de usuario

Estructura boot

El directorio `/boot` contiene los archivos necesarios para arrancar el sistema, por ejemplo, los archivos del gestor de arranque GRUB.

Por lo tanto, en `/boot` se almacenan datos que se utilizan antes de que el kernel comience a ejecutar programas en modo usuario. Esto puede incluir sectores de arranque maestro guardados.

`/boot/`

Archivos estáticos
del selector de arranque

Estructura boot

Núcleo o kernel: es un software que constituye la parte más importante del sistema operativo. Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema

/boot/

Archivos estáticos
del selector de arranque

Estructura dev

/dev (dispositivos): Esta carpeta contiene los dispositivos del sistema, incluso los que no se les ha asignado (montado) un directorio, por ejemplo: micrófonos, impresoras, pendrives (memorias USB) y dispositivos especiales (por ejemplo, /dev/null). Linux trata los dispositivos como si fueran un fichero más para facilitar el flujo de la información.

/dev/

Archivos de unidades

Estructura etc

/etc (etcétera): Aquí se guardan los ficheros de configuración de los programas instalados, así como ciertos scripts que se ejecutan en el inicio del sistema. Los valores de estos ficheros de configuración pueden ser complementados o sustituidos por los ficheros de configuración de usuario que cada uno tiene en su respectivo “home” (carpeta personal).

/etc/

Configuración de sistema
de Host específico
Directorios requeridos: opt, X11, sgml, xml

Estructura etc

Ejemplos:

/etc/opt/ Archivos de configuración para los programas alojados dentro del directorio /opt.

/etc/X11/ Archivos de configuración para el X Window System, versión 11.

/etc/

Configuración de sistema
de Host específico
Directorios requeridos: opt, X11, sgml, xml

Estructura lib

/lib (bibliotecas): Contiene las bibliotecas (mal conocidas como librerías) esenciales compartidas de los programas alojados, es decir, para los binarios en /bin/ y /sbin/, las bibliotecas para el núcleo, así como módulos y controladores (drivers).

/lib/

Librerías esenciales compartidas
y módulos de Kernel

Estructura opt

/opt (opcionales): Contiene paquetes de programas opcionales de aplicaciones estáticas, es decir, que pueden ser compartidas entre los usuarios. Dichas aplicaciones no guardan sus configuraciones en este directorio; de esta manera, cada usuario puede tener una configuración diferente de una misma aplicación, de manera que se comparte la aplicación pero no las configuraciones de los usuarios, las cuales se guardan en su respectivo directorio en /home.

Estructura usr

/usr: El directorio /usr es la segunda sección más importante del sistema de archivos, cuyos datos se pueden compartir y son de solo lectura. Contiene aplicaciones y archivos utilizados por los usuarios, a diferencia de las aplicaciones y archivos utilizados por el sistema.



Utilidades y aplicaciones de (Multi-)usuario
Jerarquía secundaria
Directorios requeridos: bin, include, lib, local, sbin, share

Estructura usr

Por ejemplo, las aplicaciones no fundamentales se encuentran dentro del directorio /usr/bin en lugar del directorio /bin y los binarios de administración del sistema no fundamentales se encuentran en el directorio /usr/sbin en lugar del directorio /sbin.

/usr/

Utilidades y aplicaciones de (Multi-)usuario
Jerarquía secundaria
Directorios requeridos: bin, include, lib, local, sbin, share

Estructura usr

Las bibliotecas para cada uno se encuentran dentro del directorio /usr/lib. El directorio /usr a su vez contiene otros directorios, por ejemplo /usr/share. El directorio /usr/local es donde se instalan las aplicaciones compiladas localmente de forma predeterminada, esto evita que pueda afectar al resto del sistema.



Utilidades y aplicaciones de (Multi-)usuario
Jerarquía secundaria
Directarios requeridos: bin, include, lib, local, sbin, share

Estructura usr

usr/share: Archivos compartidos como ficheros de configuración, imágenes, iconos, themes, etc.

/usr/X11R6/ Sistema X Window System, Versión 11, Release 6.
Este directorio se relaciona con el entorno gráfico



Estructura usr

/usr/src: Códigos fuente de algunas aplicaciones y del kernel Linux. Al igual que /mnt, esta carpeta es manejada por los usuarios directamente para que éstos puedan guardar en él el código fuente de programas y bibliotecas y así puedan accesarlo fácilmente, sin problemas con permisos. Permite que el código fuente tenga un espacio propio, accesible pero apartado de todos los usuarios.



Utilidades y aplicaciones de (Multi-)usuario
Jerarquía secundaria
Directorio requerido: bin, include, lib, local, sbin, share

Estructura var

/var (variables): Archivos variables, tales como logs, archivos spool, bases de datos, archivos de e-mail temporales, y algunos archivos temporales en general. Generalmente actúa como un registro del sistema. Ayuda a encontrar los orígenes de un problema.

/var/

Variables de archivo

Estructura var

/var/crash/ Se depositan datos e información, referentes a las caídas o errores del sistema operativo. Es más específico que /var en general.

/var/lib: Información sobre el estado actual de las aplicaciones, modificable por las propias aplicaciones.

/var/

Variables de archivo

Estructura var

/var/log: Es uno de los subdirectorios más importantes ya que aquí se guardan todo tipo de logs del sistema.

/var/

Variables de archivo

Parámetros

Los comandos de la Shell de Linux pueden recibir parámetros que afectan a la manera de ejecutarse:
comando parámetro1 parámetro2 ... parámetroN.

Los parámetros son valores o argumentos que se pueden pasar a los comandos o scripts para modificar su comportamiento o salida.

Permiten a los usuarios interactuar con los comandos y scripts de manera más dinámica y flexible.

Por ejemplo, en el comando `ls -l /home/usuario`

`-l` es un parámetro que le indica al comando `ls` que liste los archivos y directorios con información detallada, y
`/home/usuario` es un parámetro que especifica el directorio cuyo contenido se debe listar.

Características generales: Modificadores

Además de parámetros, los comandos pueden recibir modificadores propios, que van siempre precedidos de los símbolos “-” y “--”.

El primero se utiliza para indicar modificadores cuyo nombre es un único carácter, mientras que el segundo precede siempre a nombres completos de modificadores.

Los modificadores, también conocidos como opciones o flags, son parámetros especiales que se utilizan para cambiar el comportamiento de los comandos

Modificadores

Ejemplo con el comando ls:

ls -l: Muestra información detallada sobre los archivos, como permisos, propietario, tamaño y fecha de modificación.

ls -a: Muestra todos los archivos, incluidos los archivos ocultos.

ls -h: Muestra los tamaños de los archivos en formato legible por humanos (por ejemplo, "10K" para 10 kilobytes).

ls -t: Ordena los archivos por fecha de modificación, con los archivos más recientes al principio.

Modificadores

Ejemplo con el comando **ls** pero con el doble guión ‘--’:

ls --all: Es equivalente a **ls -a**, muestra todos los archivos, incluidos los archivos ocultos.

ls --color: Muestra los nombres de los archivos con diferentes colores para indicar su tipo.

Modificadores

Asimismo, es habitual que coexistan ambas versiones de un mismo modificador. Por ejemplo, un buen número de comandos ofrecen estos dos modificadores, que hacen lo mismo: -v y --verbose (hacen que la salida del comando contenga información adicional). Se podría utilizar uno u otro, indistintamente.

Ejemplo `grep -i --color=auto 'ciclo' archivo.txt`

En este caso, se utilizan los siguientes modificadores:

-i: Es el modificador corto que indica que la búsqueda no es sensible a mayúsculas/minúsculas

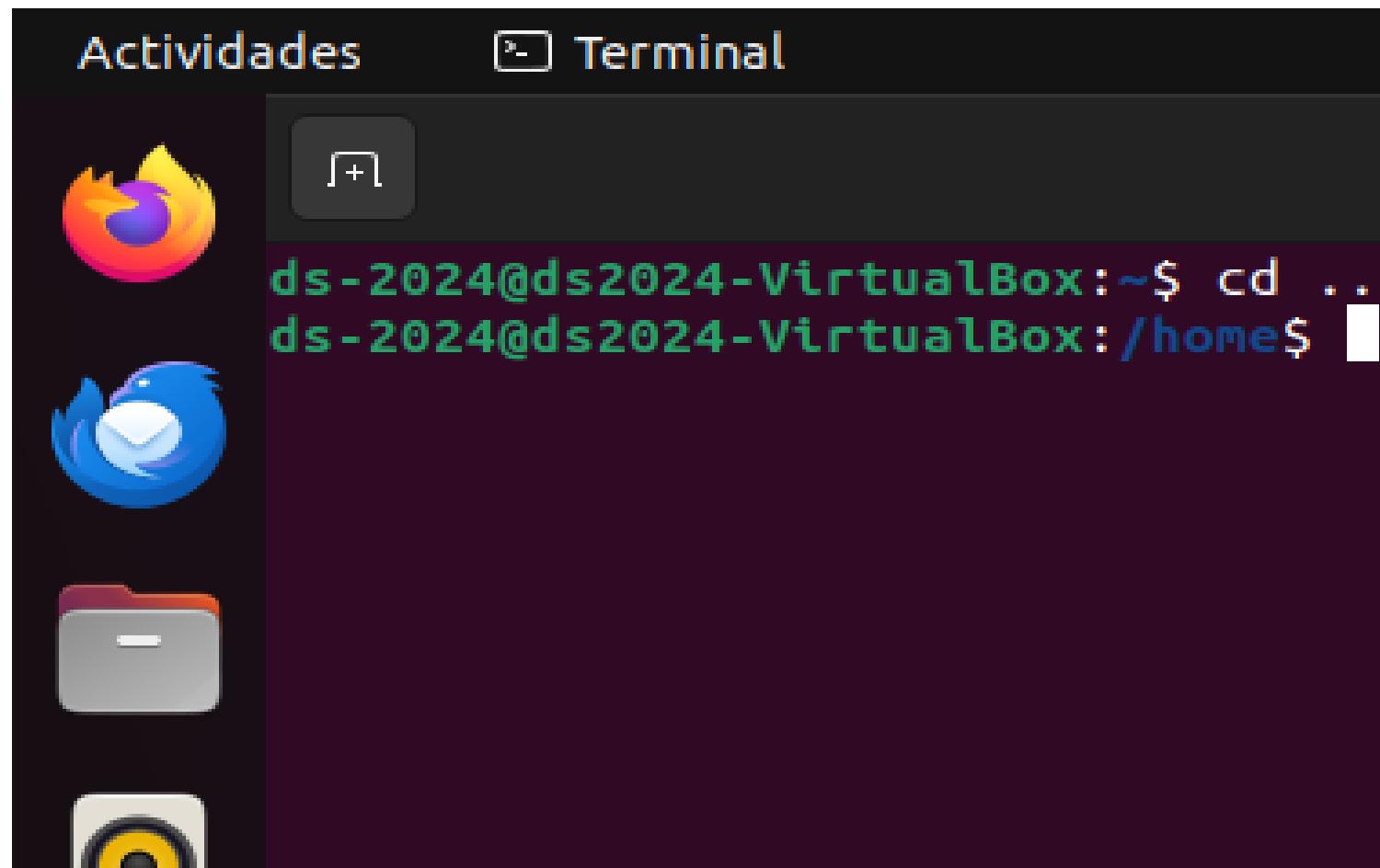
--color=auto: Es el modificador largo que resalta los resultados coincidentes en color

Actividad 5.

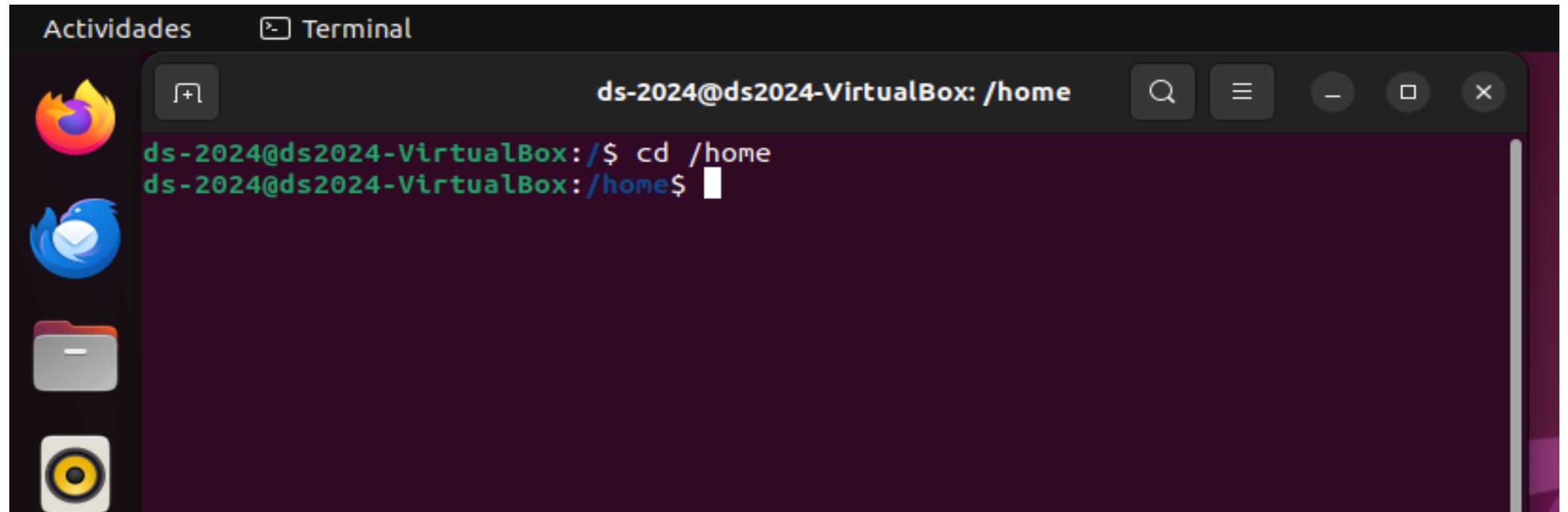
A partir de la siguiente diapositiva hasta la página 117 debe repetir cada uno de los comandos que se explican con tus propios ficheros. Una vez terminado exportar el histórico de comandos utilizado en un fichero .txt u otro similar.

Comandos básicos de navegación por el sistema de archivos

Cambio de directorio: `cd ..`

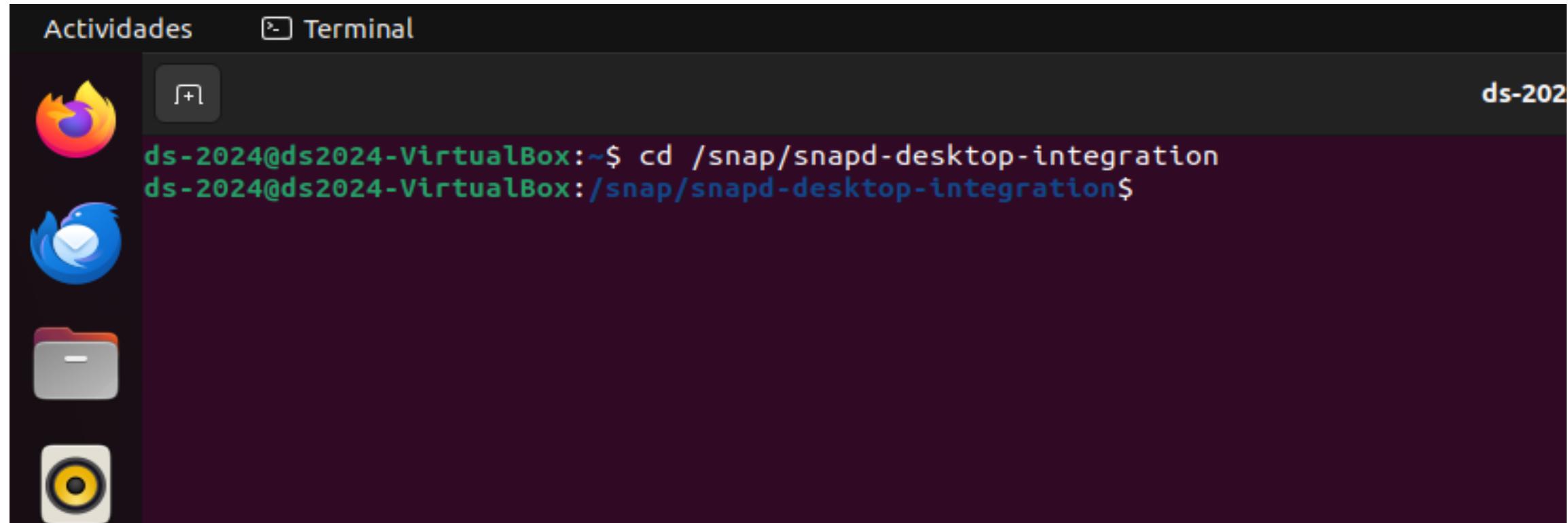


También se puede escribir cd con la ruta a llegar



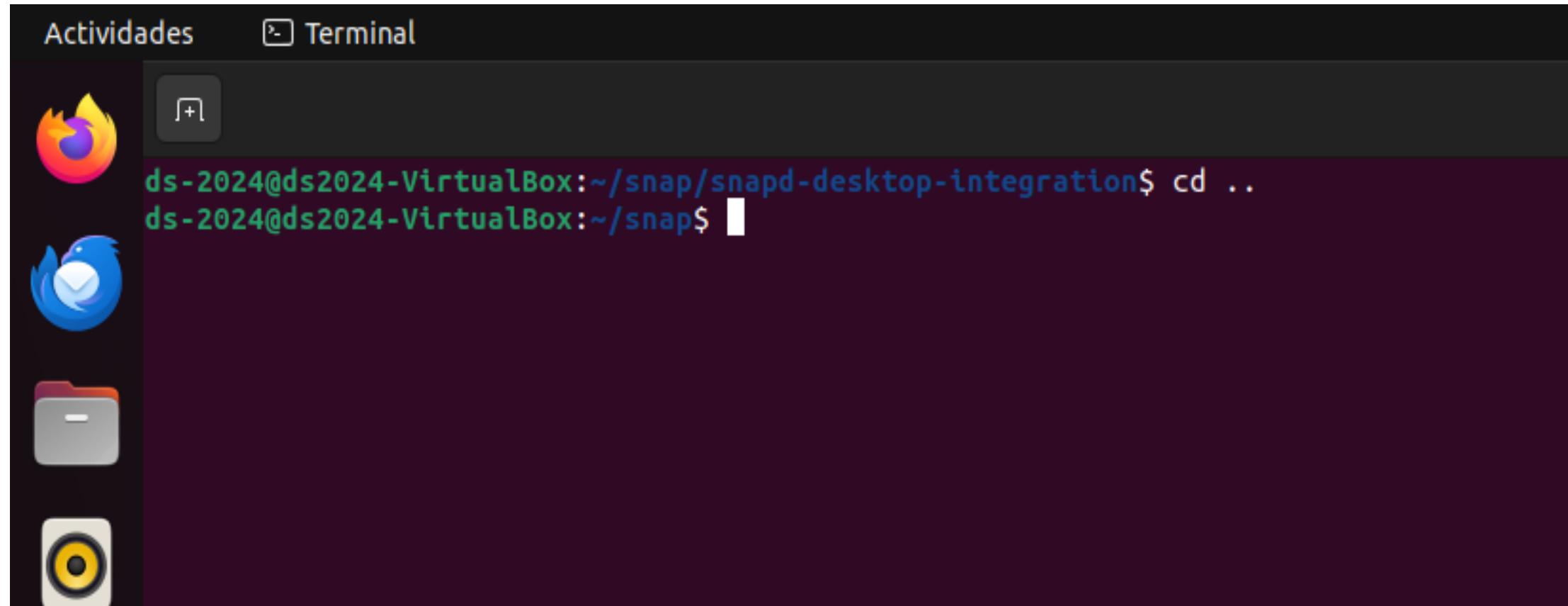
También se puede escribir cd con la ruta a llegar.

Otro ejemplo:



Asimismo, dentro de cada directorio existen dos directorios especiales: “.” y “..”

“.” hace referencia al directorio actual, mientras que
“..” se refiere al directorio padre del actual.

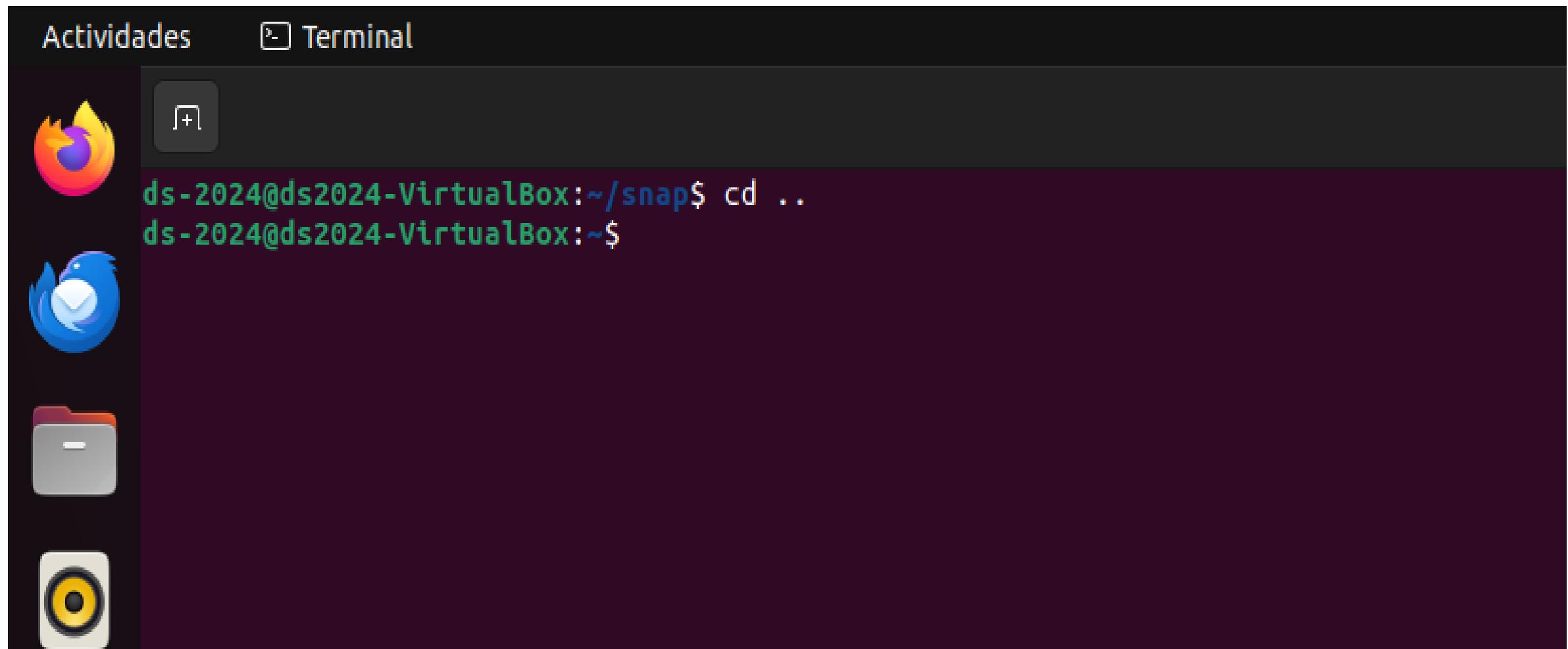
A screenshot of a Linux desktop environment. On the left, there's a dock with icons for a web browser (Firefox), a file manager (Nautilus), and other applications. In the center, a terminal window is open with the following text:

```
ds-2024@ds2024-VirtualBox:~/snap/snapd-desktop-integration$ cd ..  
ds-2024@ds2024-VirtualBox:~/snap$
```

The terminal window has a dark background and light-colored text. The desktop background is also dark.

Vuelve a escribir cd ..

Actividades Terminal

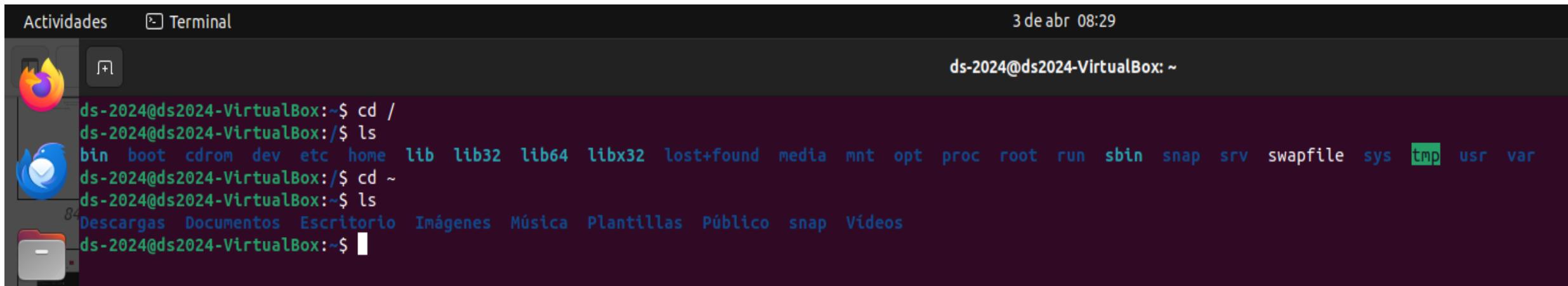


The image shows the Ubuntu Dash interface. On the left, there is a vertical sidebar with icons for various applications: a yellow fox (Firefox), a blue bird (Thunderbird), a grey folder (File Manager), and a speaker (Sound). To the right of the sidebar is a dark purple terminal window titled "Terminal". The terminal window has a title bar with the word "Terminal" and a close button. The main area of the terminal shows the following command-line session:

```
ds-2024@ds2024-VirtualBox:~/snap$ cd ..  
ds-2024@ds2024-VirtualBox:~$
```

cd ~ : Me lleva al directorio home

cd / : Me lleva al directorio raíz



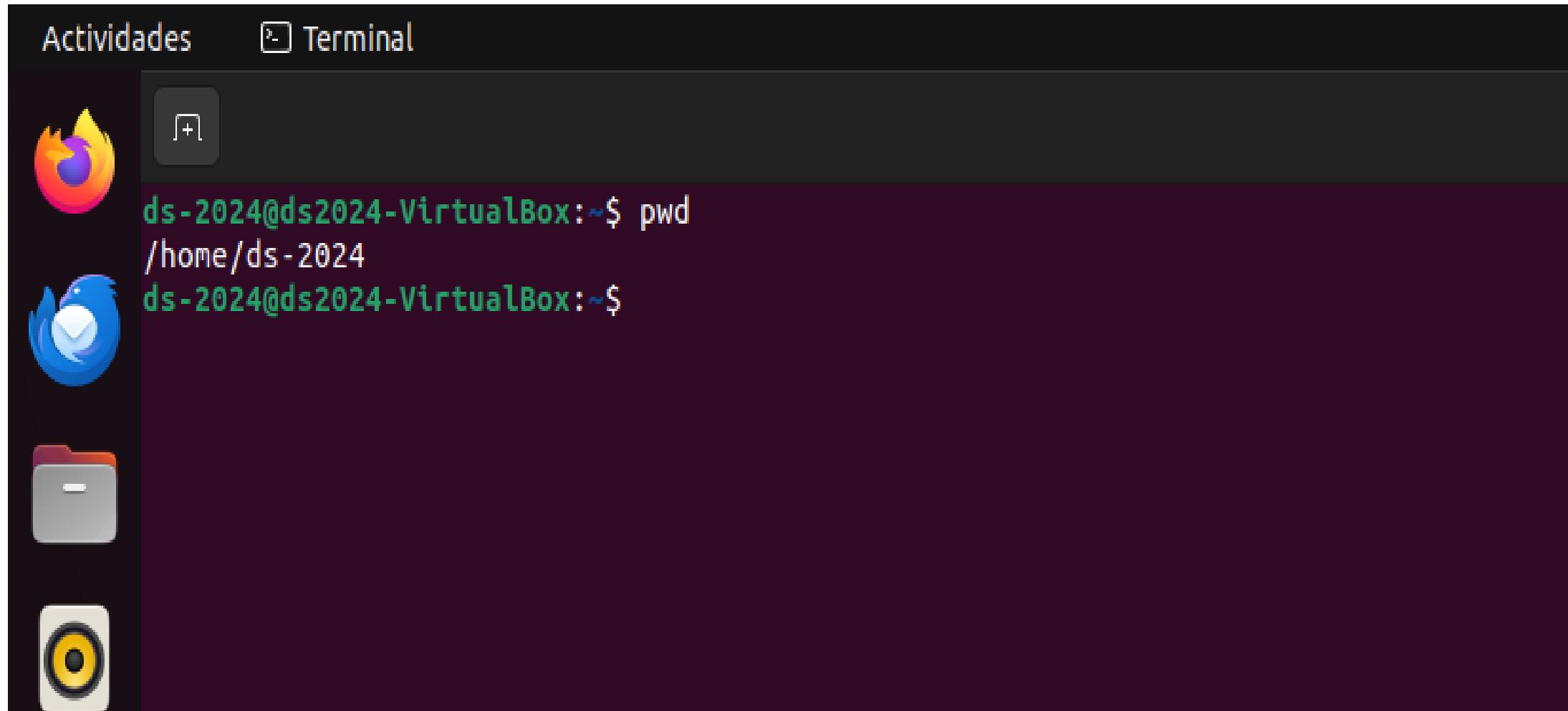
The image shows a screenshot of a Linux desktop environment, likely Ubuntu, with a dark theme. At the top, there is a dock with icons for the Dash (home), Terminal (blue square with white text), and a file browser (Nautilus). The main window is a terminal window titled "Terminal". The terminal window has a dark background and light-colored text. It displays the following command-line session:

```
Actividades Terminal 3 de abr 08:29
ds-2024@ds2024-VirtualBox:~$ cd /
ds-2024@ds2024-VirtualBox:/$ ls
bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found media mnt opt proc root run sbin snap srv swapfile sys tmp usr var
ds-2024@ds2024-VirtualBox:/$ cd ~
ds-2024@ds2024-VirtualBox:~$ ls
Descargas Documentos Escritorio Imágenes Música Plantillas Público snap Vídeos
ds-2024@ds2024-VirtualBox:~$ █
```

Para saber mi ubicación actual : **pwd**

El comando **pwd** imprime la ruta del directorio actual

pwd es la abreviación de: print working directory

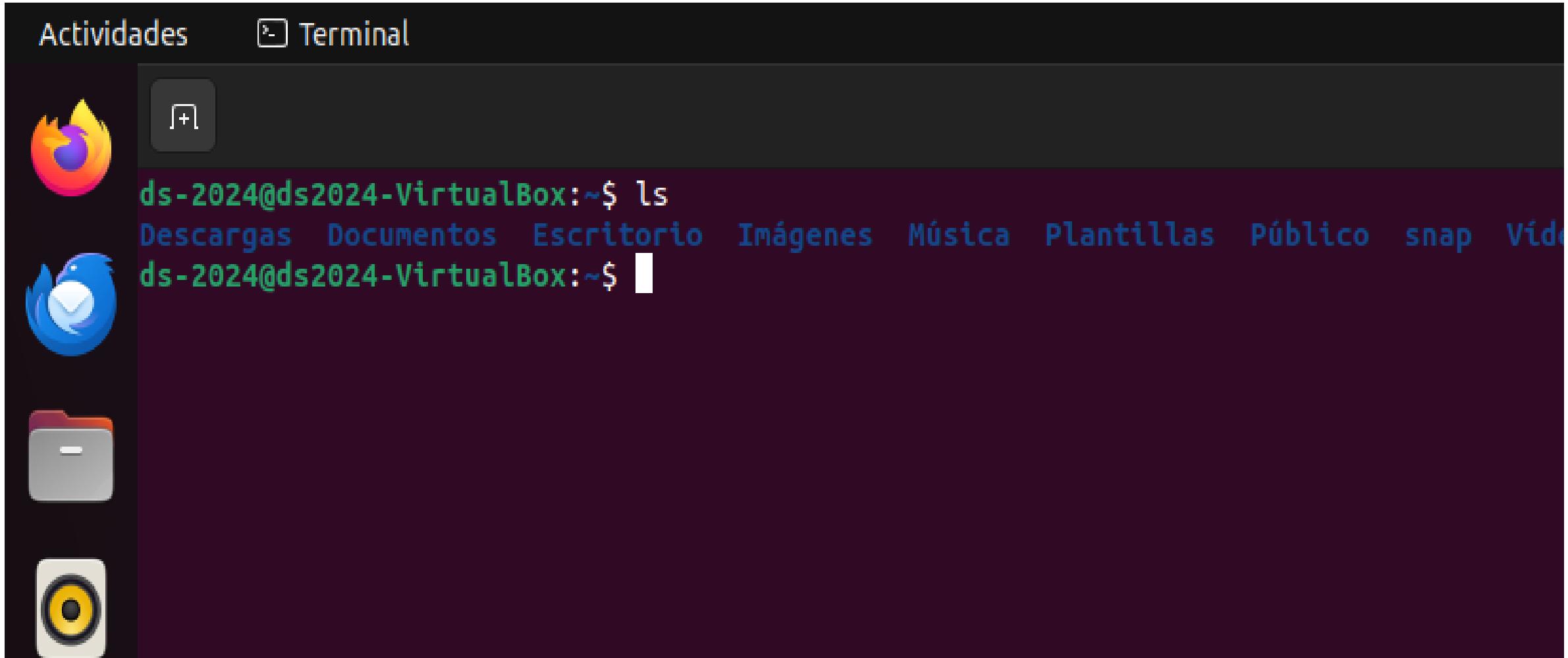


The screenshot shows a Linux desktop interface with a dark theme. At the top, there is a dock with several icons: a red and orange flame (Firefox), a blue bird (Thunderbird), a folder (File Manager), and a speaker (Speaker). To the right of the dock is a terminal window titled "Terminal". The terminal has a dark background and displays the following text:

```
ds-2024@ds2024-VirtualBox:~$ pwd
/home/ds-2024
ds-2024@ds2024-VirtualBox:~$
```

Listado del contenido de un directorio : ls

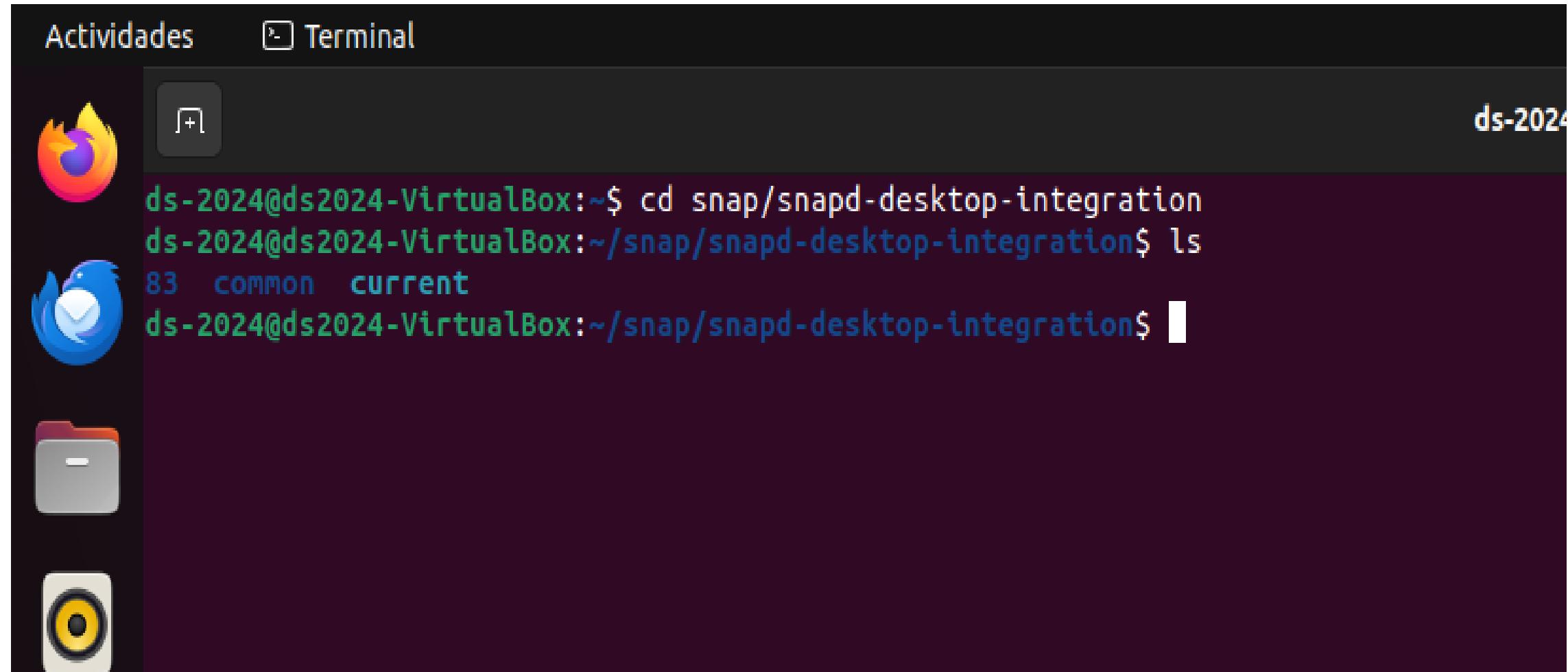
El comando ls muestra los nombres de los archivos y subdirectorios contenidos en el directorio en el que se está. Solo se obtienen los nombres de los archivos, sin ninguna otra información.

A screenshot of a Linux desktop environment. At the top, there is a dark header bar with the text "Actividades" and "Terminal". The terminal window is open and displays the following command and output:

```
ds-2024@ds2024-VirtualBox:~$ ls
Descargas Documentos Escritorio Imágenes Música Plantillas Público snap Vídeos
ds-2024@ds2024-VirtualBox:~$ █
```

The desktop background is dark, and there are icons for the Dash, Home, and Sound settings in the bottom left corner. The terminal has a dark purple background and white text.

Otro ejemplo:

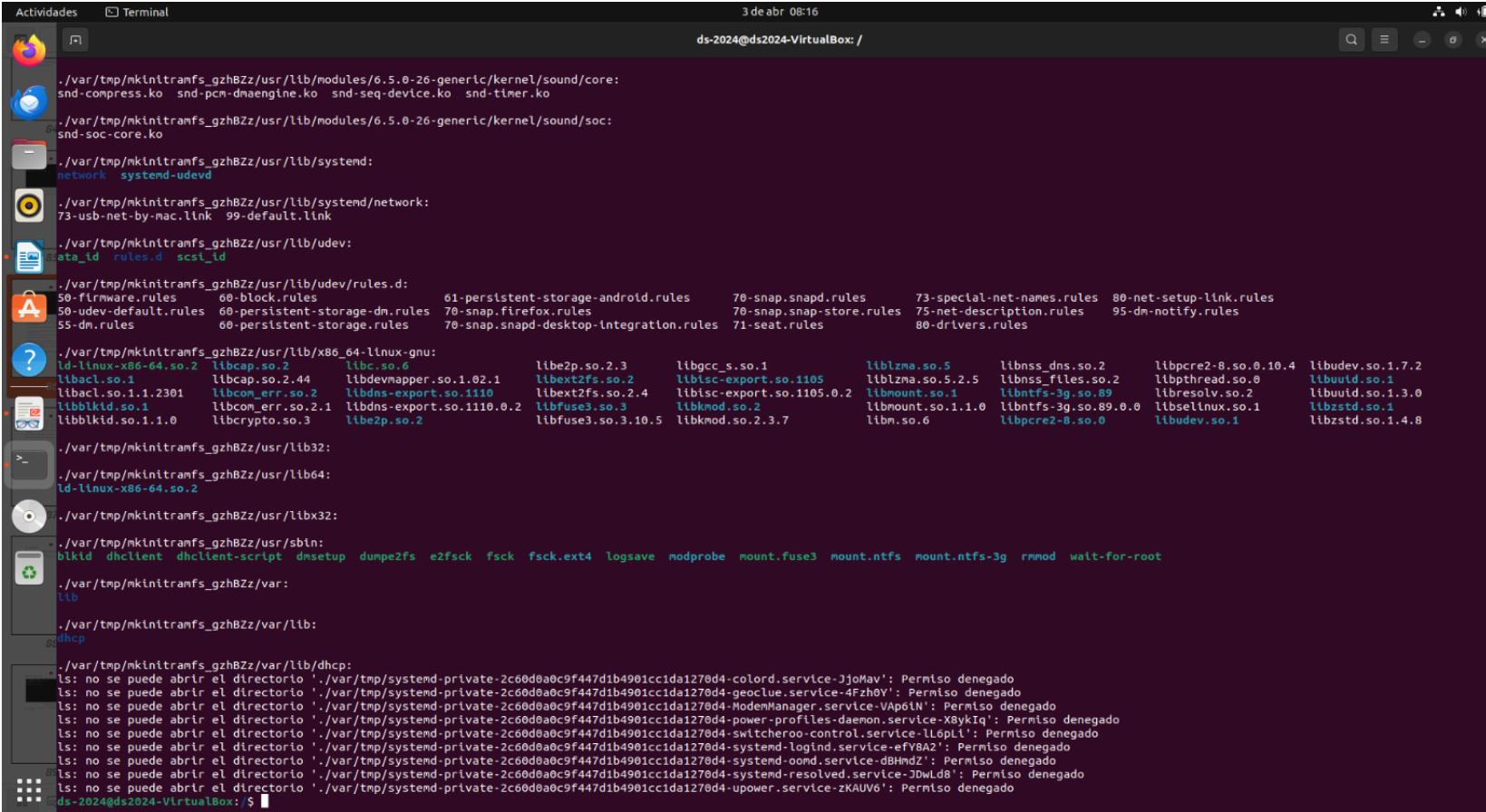


The screenshot shows a dark-themed desktop environment with a top bar containing "Actividades" and "Terminal". A search bar is also present. On the left, there are icons for a browser (Firefox), file manager, and system settings. The main area is a terminal window titled "ds-2024" which displays the following command-line session:

```
ds-2024@ds2024-VirtualBox:~$ cd snap/snapd-desktop-integration
ds-2024@ds2024-VirtualBox:~/snap/snapd-desktop-integration$ ls
83 common current
ds-2024@ds2024-VirtualBox:~/snap/snapd-desktop-integration$
```

El comando ls admite diferentes modificadores que alterarán la información mostrada en pantalla.

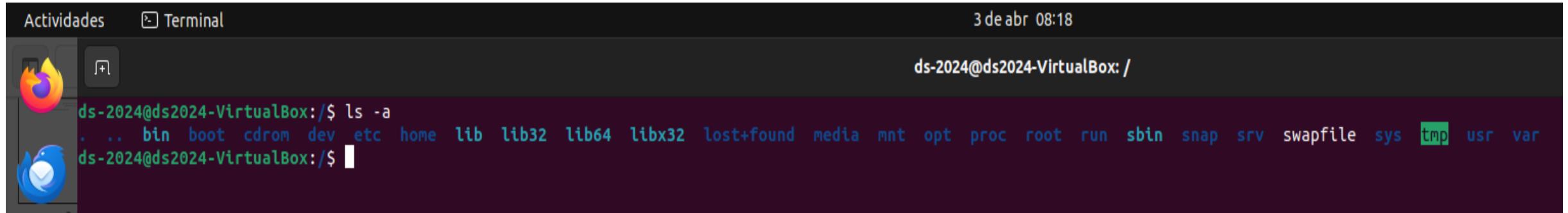
ls -R lista recursivamente los subdirectorios encontrados. Es decir, muestra los ficheros y directorios del actual, así como subdirectorios y su contenido:



The screenshot shows a terminal window titled "Terminal" with the command "ls -R" run on the root directory. The output lists all files and subdirectories in a recursive manner. The terminal interface includes a title bar with the date and time (3 de abr 08:16), a menu bar with "Actividades" and "Terminal", and a toolbar with icons for file operations. The background shows a dark desktop environment with various icons.

```
3 de abr 08:16
ds-2024@ds2024-VirtualBox: / [1]
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib/modules/6.5.0-26-generic/kernel/sound/core:
snd-compress.ko snd-pcm-dmaengine.ko snd-seq-device.ko snd-timer.ko
snd-soc-core.ko
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib/modules/6.5.0-26-generic/kernel/sound/soc:
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib/systemd:
network systemd-udevd
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib/systemd/network:
73-usb-net-by-mac.link 99-default.link
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib/udev:
data_id.rules.d scsi_id
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib/udev/rules.d:
50-firmware.rules 60-block.rules 61-persistent-storage-android.rules 70-snap.snapd.rules 73-special-net-names.rules 80-net-setup-link.rules
50-udev-default.rules 60-persistent-storage-dm.rules 70-snap.firefox.rules 70-snap.snap-store.rules 75-net-description.rules 95-dm-notify.rules
55-dm.rules 60-persistent-storage.rules 70-snap.snapd-desktop-integration.rules 71-seat.rules 80-drivers.rules
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib/x86_64-linux-gnu:
ld-linux-x86-64.so.2 libcap.so.6 libgcc_s.so.1 liblzlma.so.5 libnss_dns.so.2 libpcre2-8.so.0.10.4 libudev.so.1.7.2
libacl.so.1 libcap.so.2.44 libdevmapper.so.1.02.1 libext2fs.so.2 libisc-export.so.1105 liblzma.so.5.2.5 libnss_files.so.2 libpthread.so.0 libbuid.so.1
libacl.so.1.2301 libcom_err.so.2 libdns-export.so.1110 libext2fs.so.2.4 libisc-export.so.1105.0.2 libmount.so.1 libntfs-3g.so.89 libresolv.so.2 libuvid.so.1.3.0
libblkid.so.1 libcom_err.so.2.1 libdns-export.so.1110.0.2 libfuse3.so.3 libkmod.so.2 libmount.so.1.1.0 libntfs-3g.so.89.0.0 libselinux.so.1 libzstd.so.1
libblkid.so.1.1.0 libcrypto.so.3 libe2p.so.2 libfuse3.so.3.10.5 libkmod.so.2.3.7 libm.so.6 libpcre2-8.so.0 libudev.so.1 libzstd.so.1.4.8
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib32:
.
./var/tmp/mkinitramfs_gzhBZz/usr/lib64:
ld-linux-x86-64.so.2
.
./var/tmp/mkinitramfs_gzhBZz/usr/libx32:
.
./var/tmp/mkinitramfs_gzhBZz/usr/sbin:
blkid dhclient_dhclient-script dmsetup dumpe2fs e2fsck fsck fsck.ext4 logsave modprobe mount.fuse3 mount.ntfs mount.ntfs-3g rmmod wait-for-root
.
./var/tmp/mkinitramfs_gzhBZz/var:
lib
.
./var/tmp/mkinitramfs_gzhBZz/var/lib:
dhclient
.
./var/tmp/mkinitramfs_gzhBZz/var/lib/dhcp:
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-colord.service-JjoMav': Permisos denegados
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-geoclue.service-4Fzh0Y': Permisos denegados
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-ModemManager.service-Vap0tN': Permisos denegados
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-power-profiles-daemon.service-XBykIq': Permisos denegados
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-switcheroo-control.service-L6pL1': Permisos denegados
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-systemd-logind.service-eFyBA2': Permisos denegados
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-systemd-logind.service-eFyBA2': Permisos denegados
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-systemd-resolved.service-JDwId8': Permisos denegados
ls: no se puede abrir el directorio './var/tmp/systemd-private-2c60d0a0c9f447dib4901cc1da1270d4-upower.service-zKAUV6': Permisos denegados
.
ds-2024@ds2024-VirtualBox: $
```

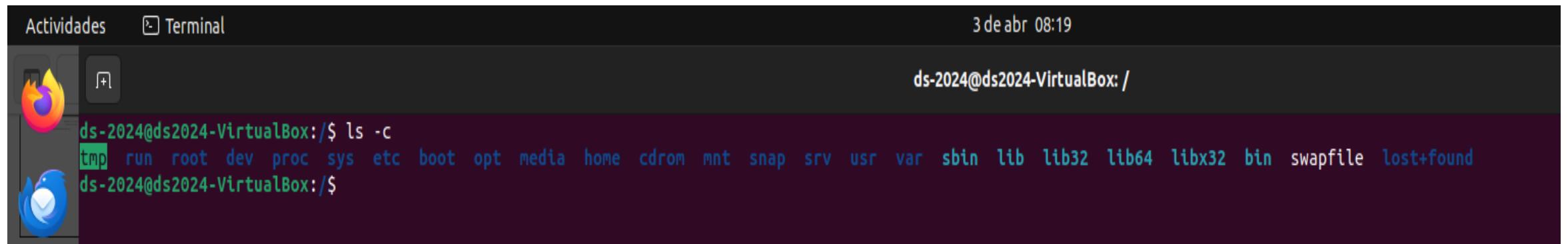
ls -a muestra todos los archivos, incluyendo algunos que están ocultos para el usuario , son aquellos que comienzan por un punto.



A screenshot of a Linux desktop environment. At the top, there's a dark header bar with the word "Actividades" and a "Terminal" icon. On the right side of the header, it shows the date "3 de abr 08:18". Below the header is a terminal window with a dark background and light-colored text. The terminal window has a title bar that says "ds-2024@ds2024-VirtualBox: /". Inside the terminal, the command "ls -a" is run, displaying a long list of directory entries including ". . ." and many files starting with a dot, such as ".bin", ".boot", ".cdrom", ".dev", ".etc", ".home", ".lib", ".lib32", ".lib64", ".libx32", ".lost+found", ".media", ".mnt", ".opt", ".proc", ".root", ".run", ".sbin", ".snap", ".srv", ".swapfile", ".sys", ".tmp", ".usr", and ".var".

```
Actividades Terminal 3 de abr 08:18
ds-2024@ds2024-VirtualBox: ~
. . .
.. bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found media mnt opt proc root run sbin snap srv swapfile sys tmp usr var
ds-2024@ds2024-VirtualBox: ~
```

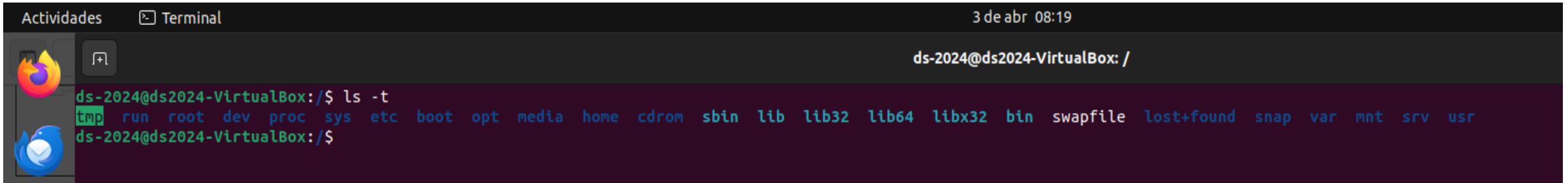
ls -c muestra el contenido del directorio ordenado por día y hora de creación.



A screenshot of a Linux desktop environment, similar to the one above. It shows a terminal window with the date "3 de abr 08:19" at the top right. The terminal window title is "ds-2024@ds2024-VirtualBox: /". The output of the "ls -c" command is shown, listing the same directory contents as the previous screenshot, but they are ordered by creation time. The output includes ".tmp", ".run", ".root", ".dev", ".proc", ".sys", ".etc", ".boot", ".opt", ".media", ".home", ".cdrom", ".mnt", ".snap", ".srv", ".var", ".sbin", ".lib", ".lib32", ".lib64", ".libx32", ".bin", ".swapfile", and ".lost+found".

```
Actividades Terminal 3 de abr 08:19
ds-2024@ds2024-VirtualBox: ~
. tmp run root dev proc sys etc boot opt media home cdrom mnt snap srv usr var sbin lib lib32 lib64 libx32 bin swapfile lost+found
ds-2024@ds2024-VirtualBox: ~
```

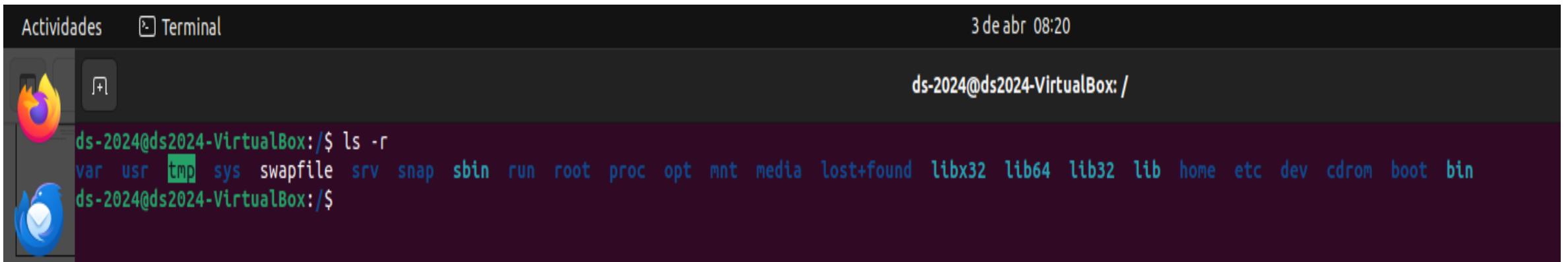
ls -t muestra el contenido del directorio ordenado por día y hora de modificación.



A screenshot of a Linux desktop environment. At the top, there's a dark header bar with the text "Actividades" and "Terminal". On the right side of the header, it shows the date and time: "3 de abr 08:19". Below the header is a terminal window with a dark background. The terminal window has a title bar with the text "ds-2024@ds2024-VirtualBox: /". Inside the terminal, the command "ls -t" is run, displaying a list of directory contents sorted by modification time. The output includes: tmp, run, root, dev, proc, sys, etc, boot, opt, media, home, cdrom, sbin, lib, lib32, lib64, libx32, bin, swapfile, lost+found, snap, var, mnt, srv, usr. The cursor is visible at the end of the command line.

```
ds-2024@ds2024-VirtualBox:$ ls -t
tmp run root dev proc sys etc boot opt media home cdrom sbin lib lib32 lib64 libx32 bin swapfile lost+found snap var mnt srv usr
ds-2024@ds2024-VirtualBox:$
```

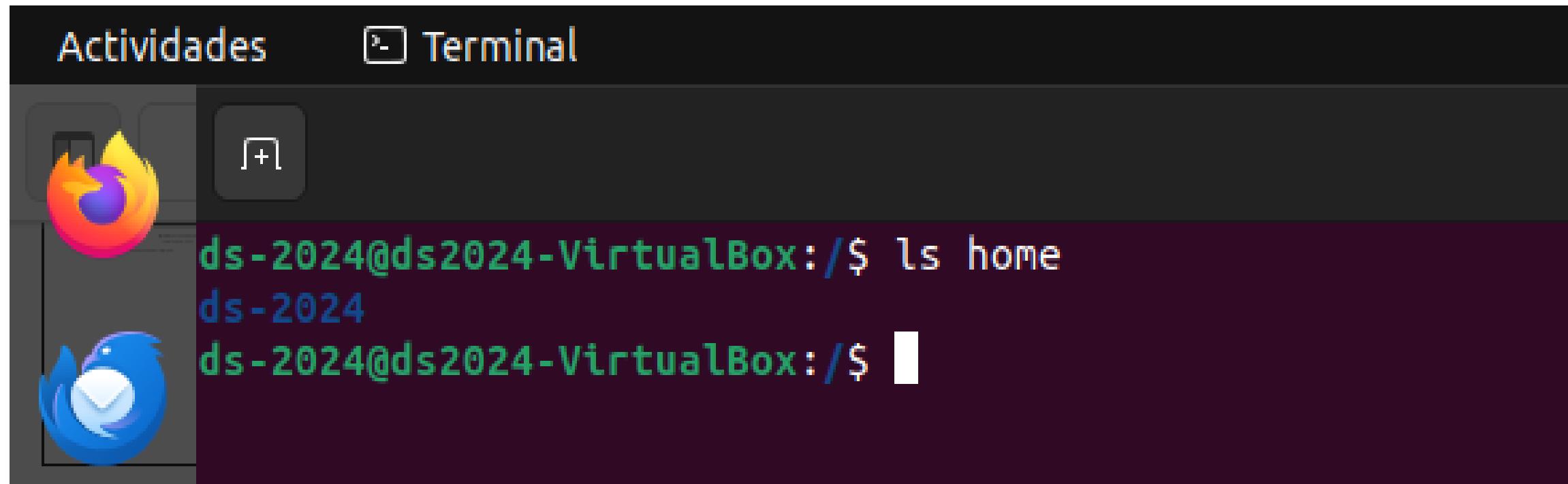
ls -r muestra el contenido del directorio y lo ordena alfabéticamente en orden inverso.



A screenshot of a Linux desktop environment, similar to the one above. It features a dark header bar with "Actividades" and "Terminal" and a date/time indicator "3 de abr 08:20". A terminal window is open with a dark background and a title bar showing "ds-2024@ds2024-VirtualBox: /". The terminal displays the command "ls -r" followed by a long list of directory contents. The list is ordered alphabetically in reverse (from z to a). The contents include: var, usr, tmp, sys, swapfile, srv, snap, sbin, run, root, proc, opt, mnt, media, lost+found, libx32, lib64, lib32, lib, home, etc, dev, cdrom, boot, bin. The cursor is at the end of the command line.

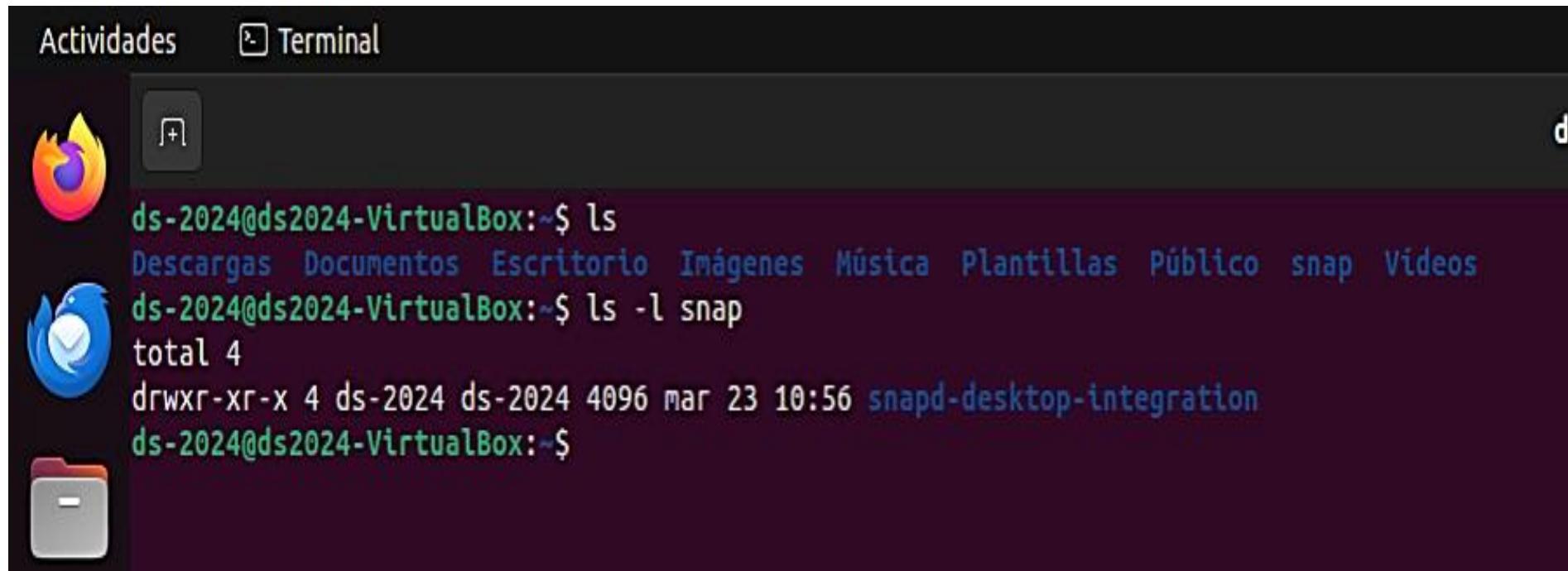
```
ds-2024@ds2024-VirtualBox:$ ls -r
var usr tmp sys swapfile srv snap sbin run root proc opt mnt media lost+found libx32 lib64 lib32 lib home etc dev cdrom boot bin
ds-2024@ds2024-VirtualBox:$
```

`ls subdir` muestra el contenido del subdirectorio “subdir”.

A screenshot of a dark-themed Ubuntu desktop environment. At the top, there is a dock with icons for the Dash, Home, and Terminal. The Terminal window is open and shows the command `ls home` being run, with the output showing a single directory named `ds-2024`.

```
ds-2024@ds2024-VirtualBox:/$ ls home
ds-2024
ds-2024@ds2024-VirtualBox:/$
```

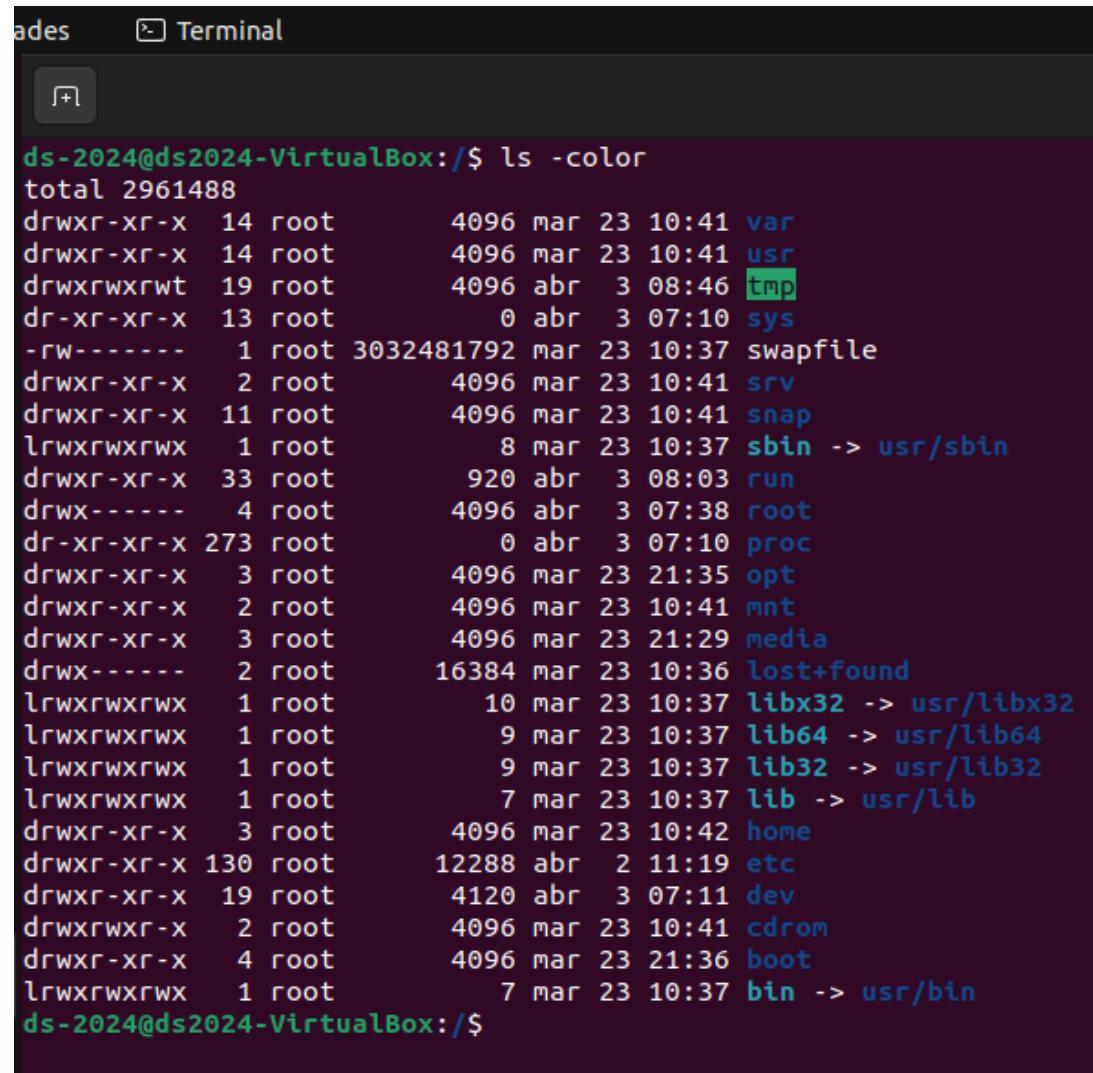
ls -l filename muestra toda la información sobre el archivo “filename”, también puede usarse para subdirectorios.



The screenshot shows a dark-themed desktop environment with a dock at the bottom containing icons for the Dash, Home, and a few other applications. A terminal window is open in the center, displaying the following command-line session:

```
Actividades Terminal
ds-2024@ds2024-VirtualBox:~$ ls
Descargas Documentos Escritorio Imágenes Música Plantillas Público snap Videos
ds-2024@ds2024-VirtualBox:~$ ls -l snap
total 4
drwxr-xr-x 4 ds-2024 ds-2024 4096 mar 23 10:56 snapd-desktop-integration
ds-2024@ds2024-VirtualBox:~$
```

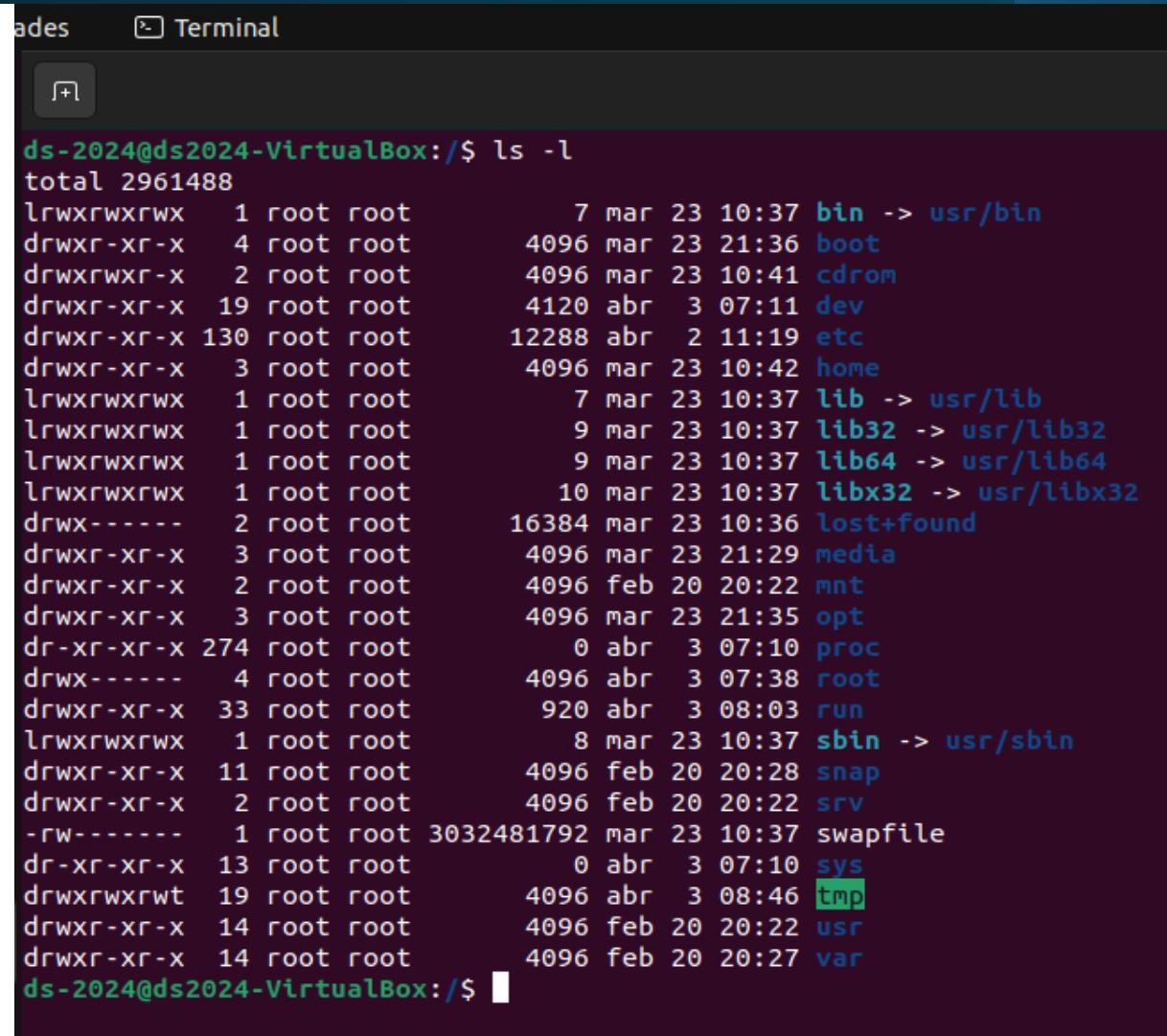
ls --color muestra el contenido del directorio coloreado: verde para los ejecutables, azul para las carpetas, fucsia para las imágenes, rojo para los comprimidos, etc.



The screenshot shows a terminal window titled "Terminal" with the command "ls -color" run by the user "ds-2024" on a "VirtualBox" system. The output lists numerous system directories and files, each colored according to its type: green for executables, blue for directories, magenta for symbolic links, and cyan for regular files. The terminal window has a dark background and light-colored text, with a small icon in the top-left corner.

```
ades Terminal
ds-2024@ds2024-VirtualBox:~$ ls -color
total 2961488
drwxr-xr-x 14 root      4096 mar 23 10:41 var
drwxr-xr-x 14 root      4096 mar 23 10:41 usr
drwxrwxrwt 19 root      4096 abr  3 08:46 tmp
dr-xr-xr-x 13 root      0 abr  3 07:10 sys
-rw-----  1 root 3032481792 mar 23 10:37 swapfile
drwxr-xr-x  2 root      4096 mar 23 10:41 srv
drwxr-xr-x 11 root      4096 mar 23 10:41 snap
lrwxrwxrwx  1 root      8 mar 23 10:37 sbin -> usr/sbin
drwxr-xr-x 33 root      920 abr  3 08:03 run
drwx----- 4 root      4096 abr  3 07:38 root
dr-xr-xr-x 273 root      0 abr  3 07:10 proc
drwxr-xr-x  3 root      4096 mar 23 21:35 opt
drwxr-xr-x  2 root      4096 mar 23 10:41 mnt
drwxr-xr-x  3 root      4096 mar 23 21:29 media
drwx----- 2 root     16384 mar 23 10:36 lost+found
lrwxrwxrwx  1 root      10 mar 23 10:37 libx32 -> usr/libx32
lrwxrwxrwx  1 root      9 mar 23 10:37 lib64 -> usr/lib64
lrwxrwxrwx  1 root      9 mar 23 10:37 lib32 -> usr/lib32
lrwxrwxrwx  1 root      7 mar 23 10:37 lib -> usr/lib
drwxr-xr-x  3 root      4096 mar 23 10:42 home
drwxr-xr-x 130 root    12288 abr  2 11:19 etc
drwxr-xr-x 19 root     4120 abr  3 07:11 dev
drwxrwxr-x  2 root      4096 mar 23 10:41 cdrom
drwxr-xr-x  4 root      4096 mar 23 21:36 boot
lrwxrwxrwx  1 root      7 mar 23 10:37 bin -> usr/bin
ds-2024@ds2024-VirtualBox:~$
```

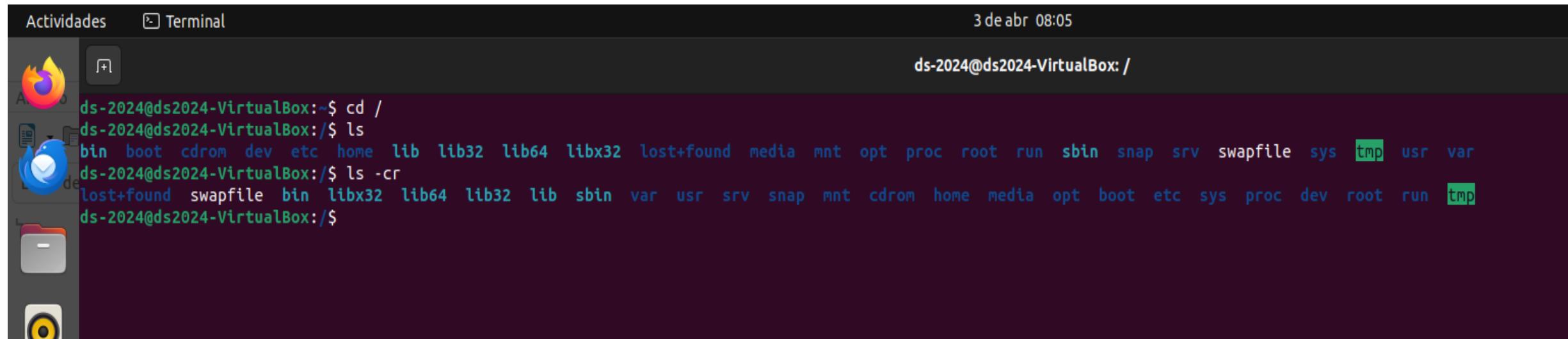
`ls -l` muestra toda la información de cada archivo, incluyendo permisos, tipo de archivo, tamaño y fecha de creación o del último cambio introducido, etc.



The screenshot shows a terminal window with the title "Terminal". The command `ls -l` is run, displaying a detailed listing of files and directories. The output includes columns for permissions, owner, group, size, date modified, and file name. Many entries are symbolic links pointing to other locations like `bin`, `boot`, `cdrom`, `dev`, `etc`, `home`, `lib`, `lib32`, `lib64`, `libx32`, `lost+found`, `media`, `mnt`, `opt`, `proc`, `root`, `run`, `sbin`, `snap`, `srv`, `swapfile`, `sys`, `tmp`, `usr`, and `var`. The file `swapfile` has a large size of `3032481792`.

```
total 2961488
lrwxrwxrwx  1 root root      7 mar 23 10:37 bin -> usr/bin
drwxr-xr-x  4 root root    4096 mar 23 21:36 boot
drwxrwxr-x  2 root root    4096 mar 23 10:41 cdrom
drwxr-xr-x 19 root root   4120 abr  3 07:11 dev
drwxr-xr-x 130 root root  12288 abr  2 11:19 etc
drwxr-xr-x  3 root root   4096 mar 23 10:42 home
lrwxrwxrwx  1 root root      7 mar 23 10:37 lib -> usr/lib
lrwxrwxrwx  1 root root     9 mar 23 10:37 lib32 -> usr/lib32
lrwxrwxrwx  1 root root     9 mar 23 10:37 lib64 -> usr/lib64
lrwxrwxrwx  1 root root    10 mar 23 10:37 libx32 -> usr/libx32
drwx----- 2 root root  16384 mar 23 10:36 lost+found
drwxr-xr-x  3 root root   4096 mar 23 21:29 media
drwxr-xr-x  2 root root   4096 feb 20 20:22 mnt
drwxr-xr-x  3 root root   4096 mar 23 21:35 opt
dr-xr-xr-x 274 root root      0 abr  3 07:10 proc
drwx----- 4 root root   4096 abr  3 07:38 root
drwxr-xr-x 33 root root    920 abr  3 08:03 run
lrwxrwxrwx  1 root root      8 mar 23 10:37 sbin -> usr/sbin
drwxr-xr-x 11 root root   4096 feb 20 20:28 snap
drwxr-xr-x  2 root root   4096 feb 20 20:22 srv
-rw-------  1 root root 3032481792 mar 23 10:37 swapfile
dr-xr-xr-x 13 root root      0 abr  3 07:10 sys
drwxrwxrwt 19 root root   4096 abr  3 08:46 tmp
drwxr-xr-x 14 root root   4096 feb 20 20:22 usr
drwxr-xr-x 14 root root   4096 feb 20 20:27 var
```

Las opciones anteriores pueden combinarse. Por ejemplo, `ls -cr` muestra el directorio ordenando inversamente por fechas



The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open in the center, displaying a command-line session. The terminal title bar says "Terminal". The date and time "3 de abr 08:05" are shown above the terminal window. The terminal window itself has a dark background with light-colored text. It contains the following commands and output:

```
Actividades Terminal 3 de abr 08:05
ds-2024@ds2024-VirtualBox: ~
ds-2024@ds2024-VirtualBox: ~$ cd /
ds-2024@ds2024-VirtualBox: /$ ls
bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found media mnt opt proc root run sbin snap srv swapfile sys tmp usr var
ds-2024@ds2024-VirtualBox: /$ ls -cr
lost+found swapfile bin libx32 lib64 lib32 lib sbin var usr srv snap mnt cdrom home media opt boot etc sys proc dev root run tmp
ds-2024@ds2024-VirtualBox: /$
```

La shell de comandos admite el uso de caracteres denominados “comodín” (o wildcards), que permiten abarcar conjuntos de ficheros o directorios cuyo nombre encaje con la expresión que definen:

El carácter “*” se utiliza para representar cualquier carácter un número indefinido de veces.

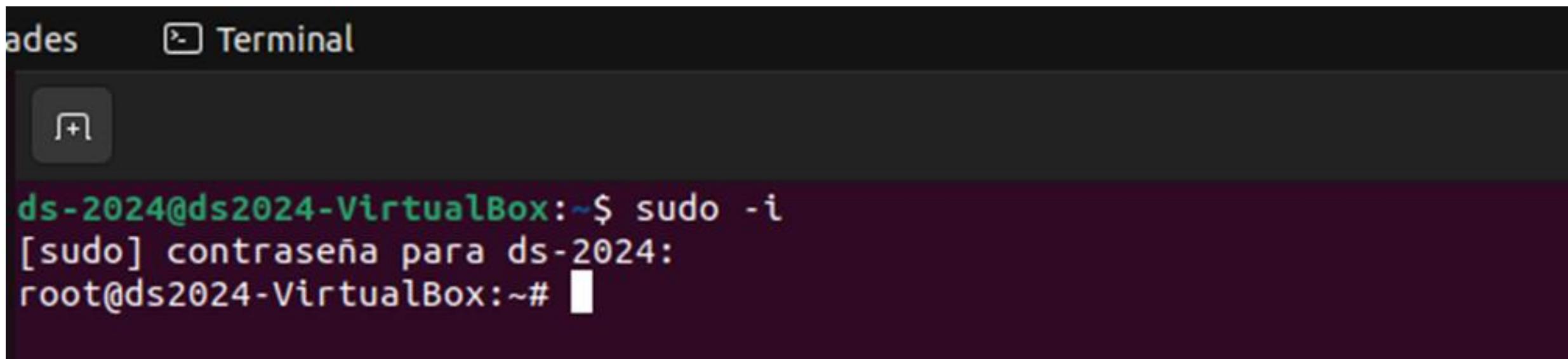
El carácter “?” se utiliza para representar cualquier carácter una única vez.

Si hay dos ficheros, “bar.txt” y “baz.txt”, se podrían listar ambos (sin considerar el resto de los ficheros del directorio) mediante el comando **ls ba?.txt**.

Asimismo, si se quisieran listar todos los ficheros con extensión .gif, podría hacerse mediante el comando **ls *.gif**.

Existe otro comando, denominado **dir**, que tiene la misma función que **ls**, pero sin mostrar tanta información.

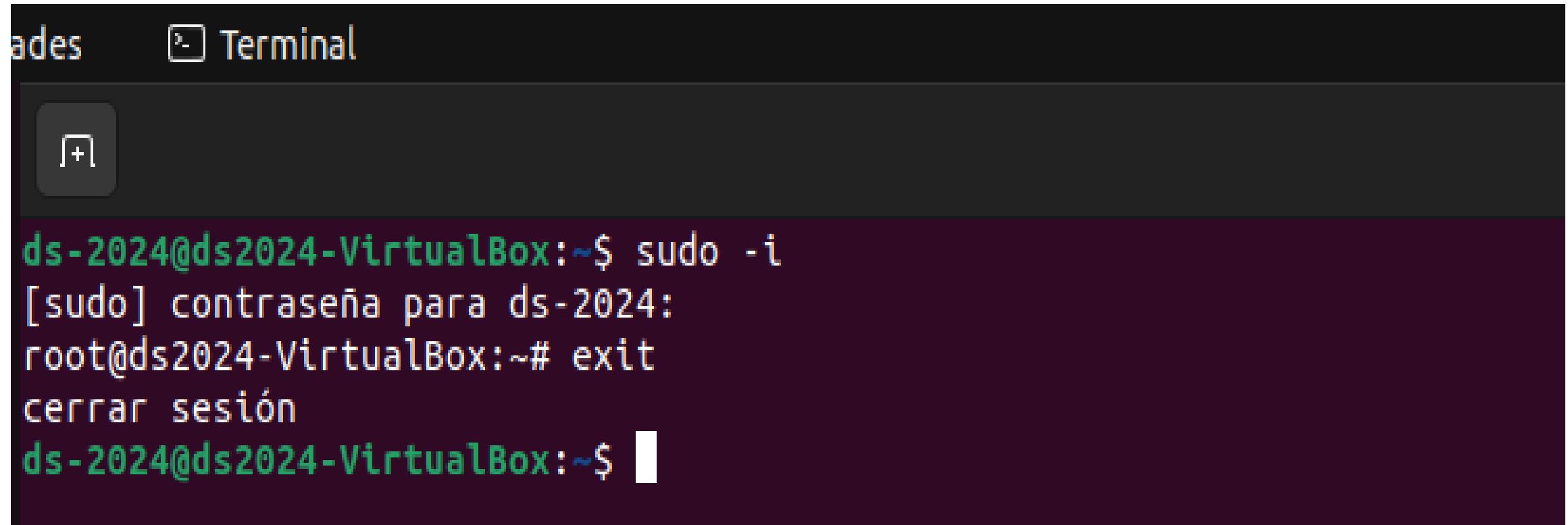
Para cambiar al modo superusuario: **sudo -i** .
Luego introducir la contraseña de root



The screenshot shows a terminal window titled "Terminal". The window has a dark background and light-colored text. At the top, there are tabs for "Ades" and "Terminal". Below the tabs, there is a small icon of a window with a plus sign. The terminal output is as follows:

```
ds-2024@ds2024-VirtualBox:~$ sudo -i
[sudo] contraseña para ds-2024:
root@ds2024-VirtualBox:~#
```

para salir del modo superusuario escribimos: **exit**



The screenshot shows a terminal window titled "Terminal". The terminal content is as follows:

```
ades Terminal
[+]
ds-2024@ds2024-VirtualBox:~$ sudo -i
[sudo] contraseña para ds-2024:
root@ds2024-VirtualBox:~# exit
cerrar sesión
ds-2024@ds2024-VirtualBox:~$
```

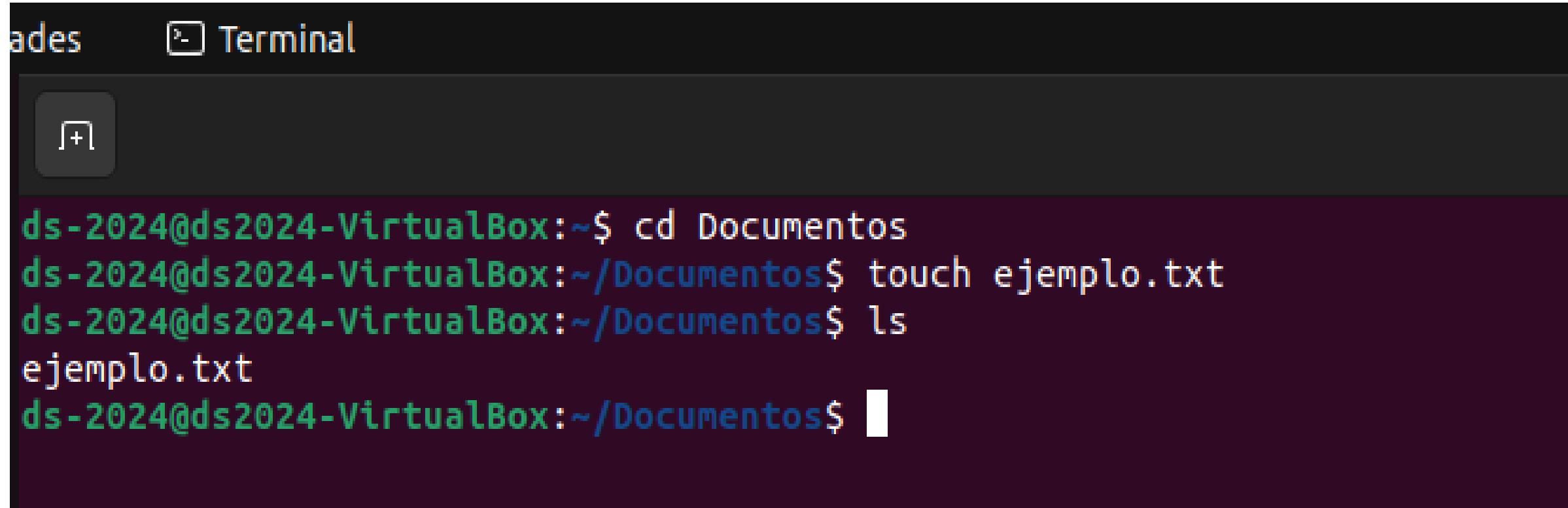
The terminal window has a dark background with light-colored text. The title bar includes the word "Terminal" and a small icon. A "+" button is visible in the top-left corner of the terminal area.

para limpiar pantalla escribimos: **clear**

Creación de ficheros

El comando **touch** actualiza los registros de fecha y hora con la fecha y hora actual de los ficheros indicados como argumento. Si el fichero no existe, el comando **touch** lo crea. Su uso más frecuente es para crear archivos.

La sintaxis del comando **touch** sigue la forma: **touch nombre_fichero**.



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there are tabs for 'Ades' and 'Terminal'. The terminal itself has a '+' icon in its title bar. The command history is as follows:

```
ds-2024@ds2024-VirtualBox:~$ cd Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ touch ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$
```

Al escribir **touch ejemplo.txt** se crea el archivo “ejemplo.txt” en el directorio actual si este no existiera. Mas ejemplos:

ades  Terminal

```
ds-2024@ds2024-VirtualBox:~$ cd Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ touch ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ touch ejemplo_2.docx
ds-2024@ds2024-VirtualBox:~/Documentos$ touch ejemplo_3.odt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
```

Creación de subdirectorios: **mkdir**

mkdir: make directory

El comando **mkdir** permite crear un nuevo subdirectorio (subcarpeta). La sintaxis es **mkdir subdir1** donde **subdir1** es el nombre del directorio (carpeta) que se va a crear

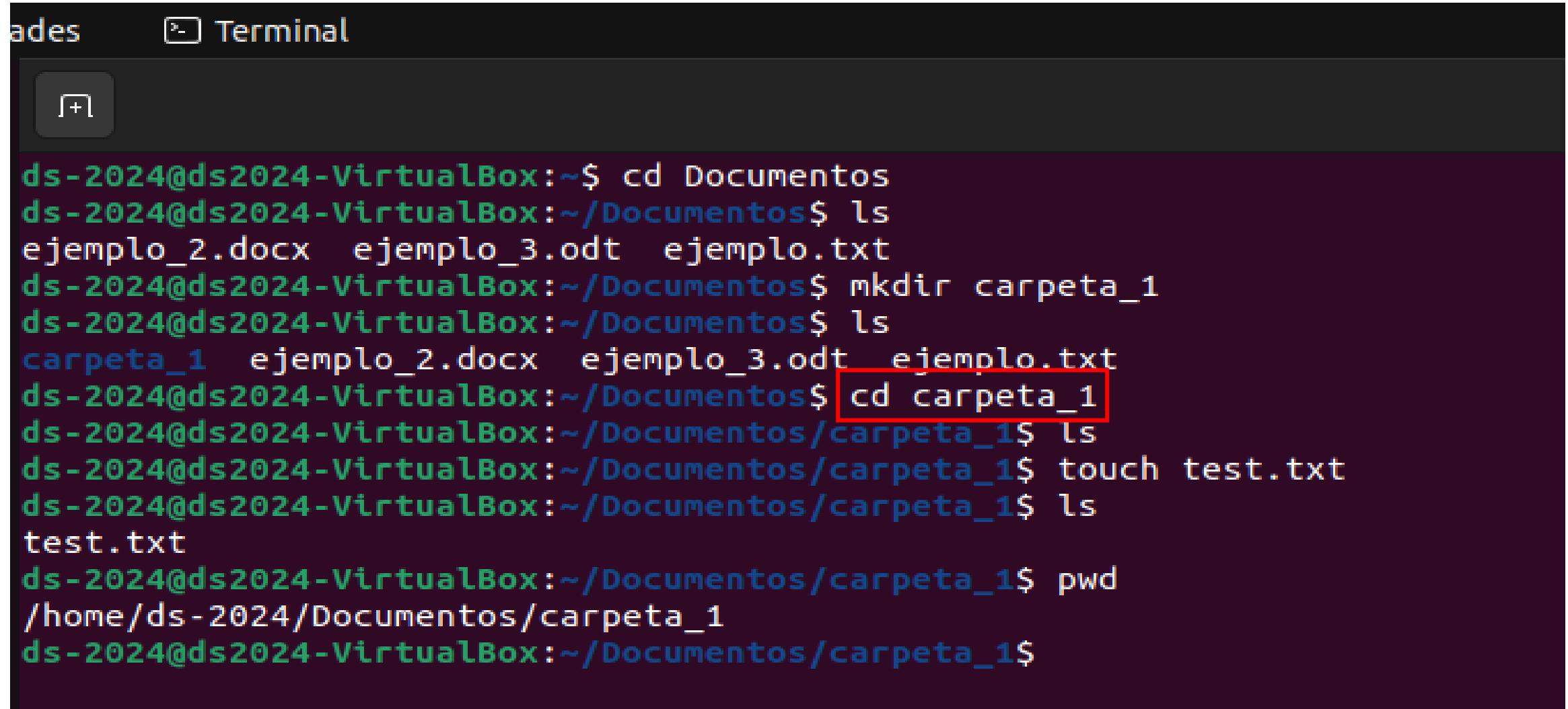
Ejemplo:

The screenshot shows a dark-themed desktop environment with a terminal window open. The terminal window has tabs for 'Ades' and 'Terminal'. A new tab icon is visible on the left. The terminal content is as follows:

```
ades      Terminal
[+]
ds-2024@ds2024-VirtualBox:~$ cd Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ mkdir carpeta_1
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_1  ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$
```

The command `mkdir carpeta_1` is highlighted with a red box.

Entramos al subdirectorio y creamos un archivo llamado test.txt



The screenshot shows a terminal window titled "Terminal" with the following session history:

```
ades Terminal
[+]
ds-2024@ds2024-VirtualBox:~$ cd Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ mkdir carpeta_1
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_1  ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ cd carpeta_1
ds-2024@ds2024-VirtualBox:~/Documentos/carpeta_1$ ls
ds-2024@ds2024-VirtualBox:~/Documentos/carpeta_1$ touch test.txt
ds-2024@ds2024-VirtualBox:~/Documentos/carpeta_1$ ls
test.txt
ds-2024@ds2024-VirtualBox:~/Documentos/carpeta_1$ pwd
/home/ds-2024/Documentos/carpeta_1
ds-2024@ds2024-VirtualBox:~/Documentos/carpeta_1$
```

The command `cd carpeta_1` is highlighted with a red box.

Borrado de subdirectorios: **rmdir**

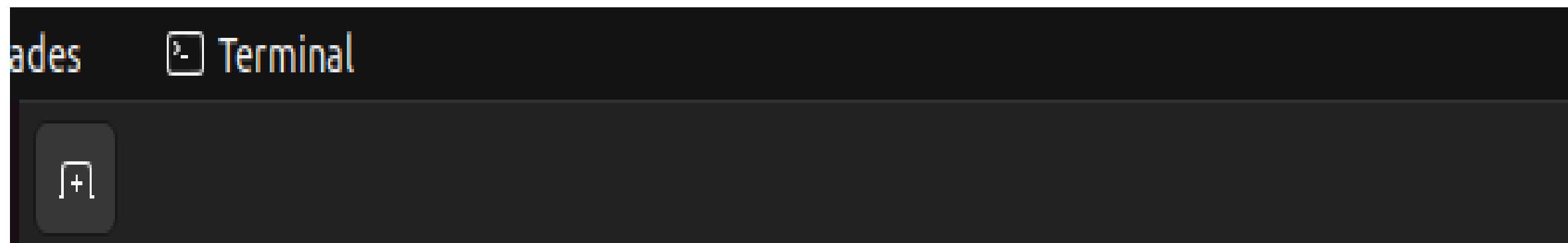
El comando **rmdir** borra uno o más directorios del sistema siempre que estos subdirectorios estén vacíos. La sintaxis es **rmdir subdir1** donde **subdir1** es el nombre del directorio que se va a eliminar.

Ejemplo:

```
ds-2024@ds2024-VirtualBox:~/Documentos$ rmdir carpeta_1
rmdir: fallo al borrar 'carpeta_1': El directorio no está vacío
ds-2024@ds2024-VirtualBox:~/Documentos$ █
```

Para borrar subdirectorios (carpetas) que contienen archivos se escribe: **rm -r nombre_directorio**

-r significa: remove



```
ades Terminal
+ ds-2024@ds2024-VirtualBox:~/Documentos$ rm -r carpeta_1
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2  ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$
```

A screenshot of a macOS desktop environment. In the top-left corner, the Dock shows the 'Finder' icon (a blue folder) and the 'Terminal' icon (a white terminal window with a blue border). The main window is a terminal session with a dark background and light-colored text. The user has run the command 'rm -r carpeta_1' to delete a directory named 'carpeta_1'. After the deletion, they run 'ls' to list the contents of the current directory, which includes 'carpeta_2', 'ejemplo_2.docx', 'ejemplo_3.odt', and 'ejemplo.txt'. The terminal prompt ends with a '\$' sign.

Copia de archivos: cp

cp: copy

Sintaxis: cp [opciones] origen destino

[opciones]: Son opciones adicionales que modifican el comportamiento del comando cp. Algunas opciones comunes son:

-v: Muestra información detallada sobre el proceso de copia.

-r: Copia directorios de forma recursiva (incluidos todos los archivos y subdirectorios).

-n: No sobrescribe archivos existentes.

-b: Crea una copia de seguridad del archivo original.

origen: Es la ubicación del archivo o directorio que deseas copiar. Puede ser un nombre de archivo, una ruta completa o un comodín.

destino: Es la ubicación donde deseas copiar el archivo o directorio. Puede ser un nombre de archivo, una ruta completa o un directorio existente.

Ejemplo

```
andes  ➔ Terminal

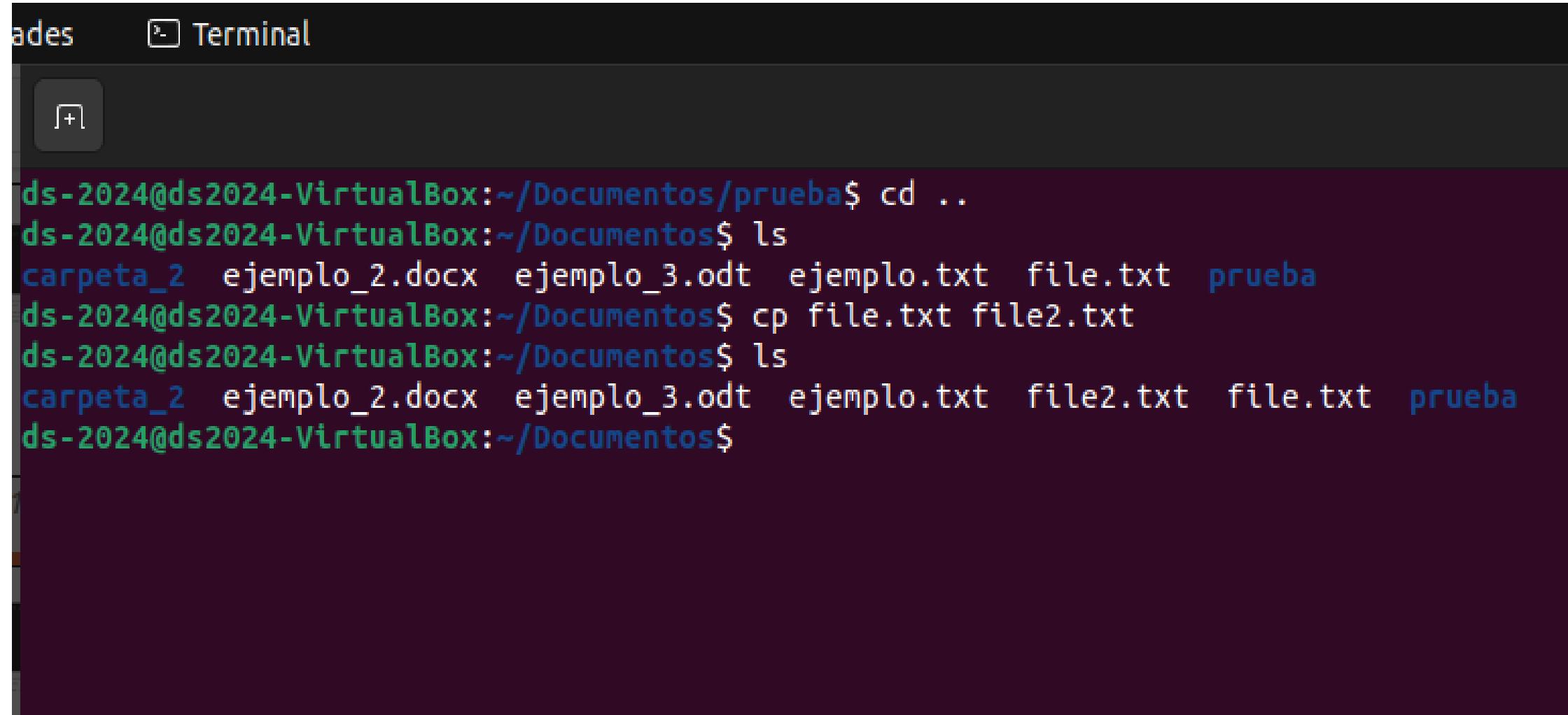
ds-2024@ds2024-VirtualBox:~/Documentos$ mkdir prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt ejemplo.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ touch file.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt ejemplo.txt file.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ cp file.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt ejemplo.txt file.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ cd prueba
ds-2024@ds2024-VirtualBox:~/Documentos/prueba$ ls
file.txt
ds-2024@ds2024-VirtualBox:~/Documentos/prueba$
```

Copiar un archivo y cambiarle el nombre:

Copia el archivo original y lo guarda con un nuevo nombre.

Sintaxis: **cp archivo.algo nuevo_archivo.algo**

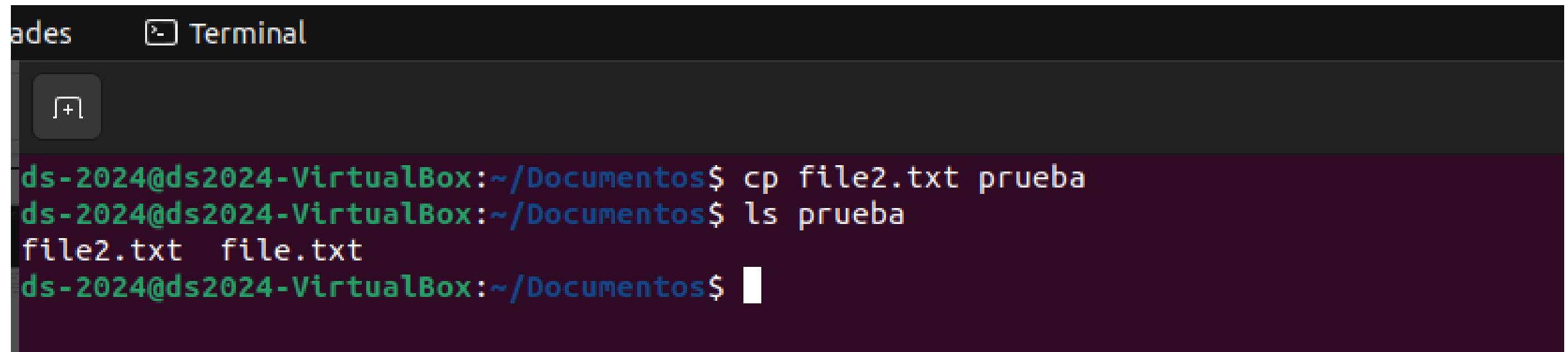
Copiar un archivo y cambiarle el nombre:



The screenshot shows a terminal window titled "Terminal" with a dark theme. The window contains the following command-line session:

```
ades Terminal
[+]
ds-2024@ds2024-VirtualBox:~/Documentos/prueba$ cd ..
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt ejemplo.txt file.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ cp file.txt file2.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt ejemplo.txt file2.txt file.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$
```

Otro ejemplo especificando ruta:



The screenshot shows a terminal window titled "Terminal" with a dark theme. The window has a header bar with tabs for "ades" and "Terminal". Below the header is a toolbar with a "+" icon. The terminal window displays the following command-line session:

```
ds-2024@ds2024-VirtualBox:~/Documentos$ cp file2.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ ls prueba
file2.txt  file.txt
ds-2024@ds2024-VirtualBox:~/Documentos$
```

¿Qué otras cosas se hacen con el comando cp? Escribir en el terminal: cp --help

```
andes Terminal ds-2024@ds2024-VirtualBox:~/Documentos$ cp --help
Modo de empleo: cp [OPCIÓN]... ORIGEN DESTINO
    o bien: cp [OPCIÓN]... ORIGEN... DIRECTORIO
    o bien: cp [OPCIÓN]... -t DIRECTORIO ORIGEN...
Copia ORIGEN a DESTINO, o varios ORIGEN(es) a DIRECTORIO.

Los argumentos obligatorios para las opciones largas son también obligatorios
para las opciones cortas.
-a, --archive           lo mismo que -dR --preserve=all
--attributes-only      no copia los datos del fichero, solamente los
                       atributos
--backup[=CONTROL]     crea una copia de seguridad de cada fichero de
                       destino que existe
-b                      como --backup pero no acepta ningún argumento
--copy-contents        copia el contenido de los ficheros especiales
                       cuando opera recursivamente
-d                      lo mismo que --no-dereference --preserve=link
-f, --force              si un fichero de destino no se puede abrir, lo
                       borra y lo intenta de nuevo (no se tiene en
                       cuenta si se utiliza también la opción -n)
-i, --interactive       pide confirmación antes de sobreescibir
-H                      sigue los enlaces simbólicos de la línea
                       de órdenes
-l, --link               crea enlaces duros de los ficheros en vez de copiarlos
-L, --dereference        siempre sigue los enlaces simbólicos en ORIGEN
-n, --no-clobber         no sobreescribe un fichero que existe
                       (tiene prioridad sobre una opción -i anterior)
-P, --no-dereference   nunca sigue los enlaces simbólicos en ORIGEN
-P, --preserve[=LISTA_ATTR] conserva si puede los atributos especificados,
                       (por omisión: mode,ownership,timestamps)
                       atributos adicionales: context, links, xattr,
                       all
--no-preserve=LISTA_ATTR no conserva los atributos especificados
--parents añade el directorio de origen a DIRECTORIO
-R, -r, --recursive     copia directorios recursivamente
--reflink[=CUÁNDO]       controla copias clonadas/Cow. Ver más abajo.
--remove-destination   borra cada fichero de destino que exista antes
                       de intentar abrirlo (compárese con --force).
--sparse=CUÁNDO         controla la creación de ficheros dispersos.
                       Véase más abajo.
--strip-trailing-slashes elimina todas las barras finales de cada
                       argumento ORIGEN
-s, --symbolic-link     crea enlaces simbólicos en lugar de copiarlos
-S, --suffix=SUFijo     reemplaza el sufijo de respaldo habitual
-t, --target-directory=DIRECTORIO copia todos los argumentos ORIGEN al
                       directorio DIRECTORIO
-T, --no-target-directory considera DEST como un archivo normal
-u, --update             copia solamente cuando el fichero ORIGEN es
                       más moderno que el fichero de destino,
                       o cuando falta el fichero de destino
-v, --verbose            da detalles sobre lo que se va haciendo
-x, --one-file-system   permanece en este sistema de ficheros
-Z, --context[=CTX]      establece el contexto de seguridad SELinux del fichero de
                       destino al tipo predeterminado
                           cosa_Z si se especifica CTX entonces establece
```

El parámetro **--help** se puede utilizar para todos los comandos:

Escribir: **ls --help**

Mover Archivos: mv origen destino

Mueve los archivos de un *directorio a otro*, o *del directorio actual a otro directorio*.

Ejemplo 1:

ades 2 Terminal



```
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt ejemplo.txt file2.txt file.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ mv ejemplo.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt file2.txt file.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ ls prueba
ejemplo.txt file2.txt file.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ 
```

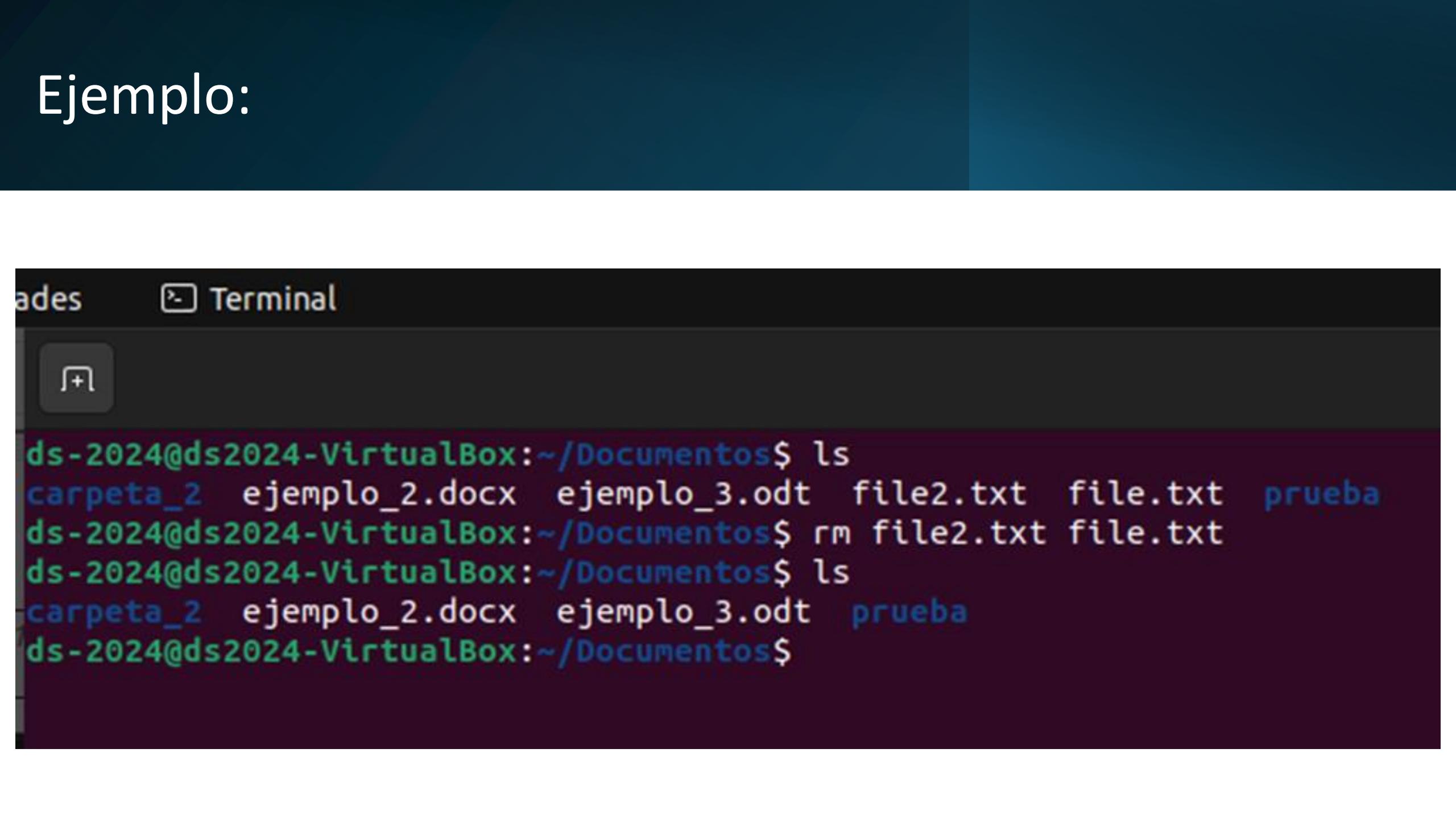
Ejemplo 2:

```
ades Terminal 3 d
+ ds-2024@ds2024-VirtualBox:~/Documentos$ touch archivo1.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
archivo1.txt  carpeta_2  ejemplo_2.docx  ejemplo_3.odt  file2.txt  file.txt  prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ mv archivo1.txt carpeta_2
ds-2024@ds2024-VirtualBox:~/Documentos$ ls carpeta_2
archivo1.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ pwd
/home/ds-2024/Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ mv /home/ds-2024/Documentos/carpeta_2/archivo1.txt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ ls prueba
archivo1.txt  ejemplo.txt  file2.txt  file.txt
ds-2024@ds2024-VirtualBox:~/Documentos$
```

Borrado de archivos: **rm**

El comando **rm** elimina uno o más archivos de un directorio en el cual se tenga permiso de escritura. La sintaxis es **rm file1 file2** . En el comando, se pueden utilizar los caracteres comodín * y ?.

Ejemplo:



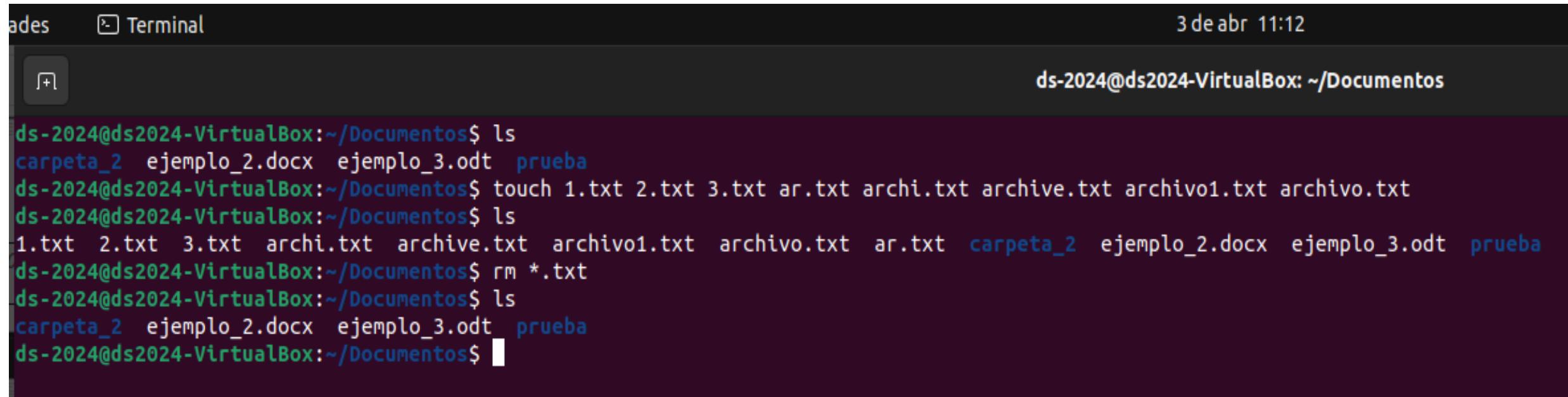
The screenshot shows a dark-themed macOS desktop environment. At the top, the Dock contains two icons: 'Alfred' (with a search field) and 'Terminal'. Below the Dock is a dark window titled 'Terminal'. The terminal window has a small '+' icon in its title bar. The main area of the terminal displays the following command-line session:

```
ades      □ Terminal

+         

ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2  ejemplo_2.docx  ejemplo_3.odt  file2.txt  file.txt  prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ rm file2.txt file.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2  ejemplo_2.docx  ejemplo_3.odt  prueba
ds-2024@ds2024-VirtualBox:~/Documentos$
```

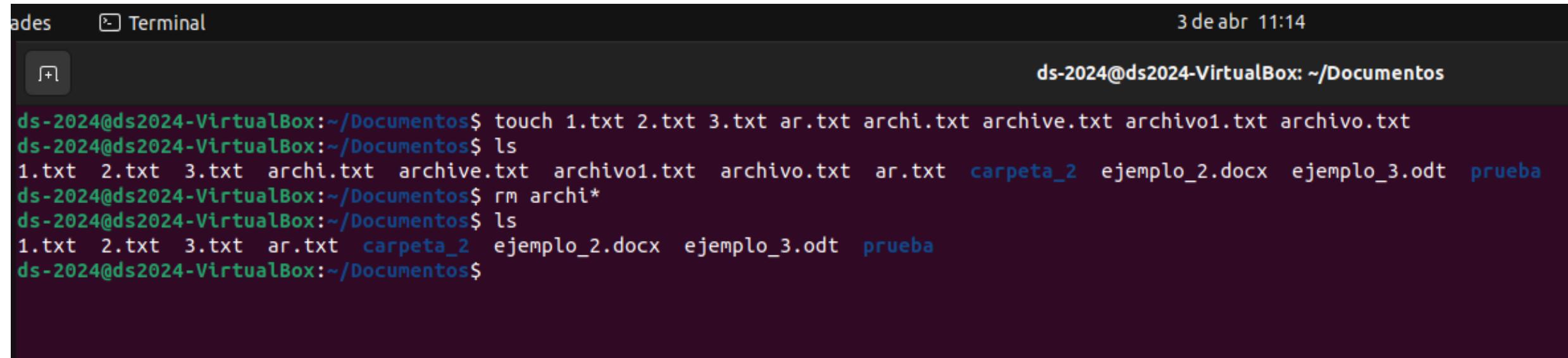
Para eliminar todos los archivos con una extensión específica: **rm *.txt**



The screenshot shows a terminal window with the following session:

```
ades Terminal 3 de abr 11:12
ds-2024@ds2024-VirtualBox: ~/Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ touch 1.txt 2.txt 3.txt ar.txt archi.txt archive.txt archivo1.txt archivo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
1.txt 2.txt 3.txt archi.txt archive.txt archivo1.txt archivo.txt ar.txt carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ rm *.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$
```

Eliminar archivos que comienzan con un nombre específico: **rm nombre***



The screenshot shows a terminal window with the following details:

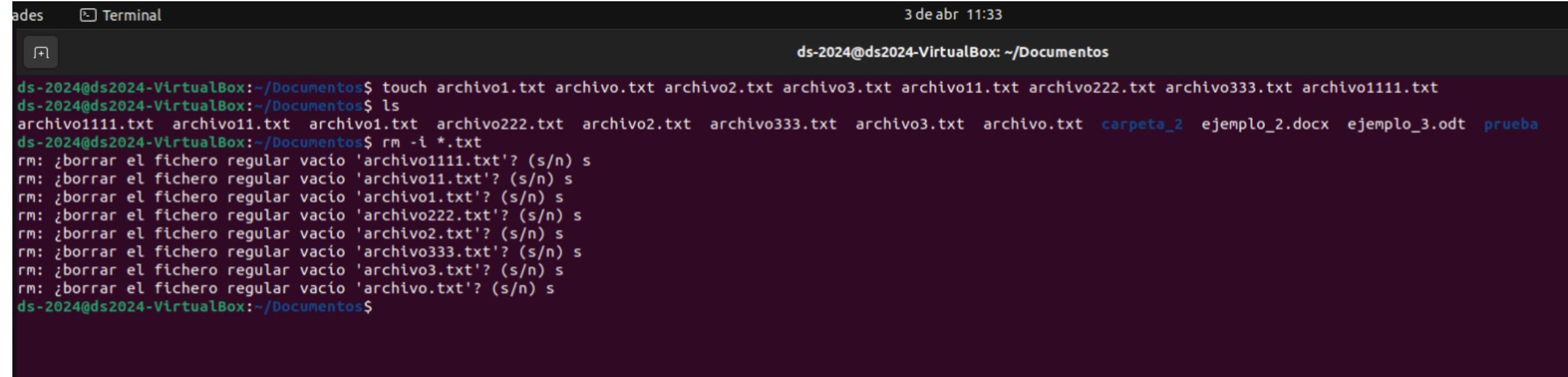
- Top bar: "ades" (partially visible), "Terminal", and the date/time "3 de abr 11:14".
- User information: "ds-2024@ds2024-VirtualBox: ~/Documentos".
- Terminal content:

```
ds-2024@ds2024-VirtualBox:~/Documentos$ touch 1.txt 2.txt 3.txt ar.txt archi.txt archive.txt archivo1.txt archivo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
1.txt 2.txt 3.txt archi.txt archive.txt archivo1.txt archivo.txt ar.txt carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ rm archi*
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
1.txt 2.txt 3.txt ar.txt carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$
```

Eliminar archivos que contienen una letra específica: rm archivo?.algo

```
ades Terminal 3 de abr 11:26
ds-2024@ds2024-VirtualBox:~/Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ touch archivo1.txt archivo.txt archivo2.txt archivo3.txt archivo11.txt archivo222.txt archivo333.txt archivo1111.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
archivo1111.txt archivo11.txt archivo1.txt archivo222.txt archivo2.txt archivo333.txt archivo3.txt archivo.txt carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ rm archivo?.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
archivo1111.txt archivo11.txt archivo222.txt archivo333.txt archivo.txt carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ rm archivo??.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
archivo1111.txt archivo222.txt archivo333.txt archivo.txt carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ rm archivo???.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
archivo1111.txt archivo.txt carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$ rm archivo????.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
archivo.txt carpeta_2 ejemplo_2.docx ejemplo_3.odt prueba
ds-2024@ds2024-VirtualBox:~/Documentos$
```

Para solicitar confirmación antes de eliminar archivos se puede escribir: **rm -i *.txt**



The screenshot shows a terminal window with the following details:

- Top bar: "ades" and "Terminal".
- Time: "3 de abr 11:33".
- User and location: "ds-2024@ds2024-VirtualBox: ~/Documentos".
- Command history:
 - "touch archivo1.txt archivo.txt archivo2.txt archivo3.txt archivo11.txt archivo222.txt archivo333.txt archivo1111.txt"
 - "ls"
 - "rm -i *.txt"
 - Confirmation loop:
 - rm: ¿borrar el fichero regular vacío 'archivo1111.txt'? (s/n) s
 - rm: ¿borrar el fichero regular vacío 'archivo11.txt'? (s/n) s
 - rm: ¿borrar el fichero regular vacío 'archivo1.txt'? (s/n) s
 - rm: ¿borrar el fichero regular vacío 'archivo222.txt'? (s/n) s
 - rm: ¿borrar el fichero regular vacío 'archivo2.txt'? (s/n) s
 - rm: ¿borrar el fichero regular vacío 'archivo333.txt'? (s/n) s
 - rm: ¿borrar el fichero regular vacío 'archivo3.txt'? (s/n) s
 - rm: ¿borrar el fichero regular vacío 'archivo.txt'? (s/n) s
 - "ds-2024@ds2024-VirtualBox:~/Documentos\$"

Eliminar todos los archivos en el directorio actual: rm *



The screenshot shows a terminal window titled "Terminal" with the following session history:

```
ades Terminal 3 de abr 11:36
ds-2024@ds2024-VirtualBox: ~/Documentos/prueba1
[+]
ds-2024@ds2024-VirtualBox:~/Documentos$ mkdir prueba1
ds-2024@ds2024-VirtualBox:~/Documentos$ cd prueba1
ds-2024@ds2024-VirtualBox:~/Documentos/prueba1$ touch archivo1.txt archivo.txt archivo2.txt archivo3.txt archivo11.txt archivo222.txt archivo333.txt archivo1111.txt
ds-2024@ds2024-VirtualBox:~/Documentos/prueba1$ ls
archivo1111.txt archivo11.txt archivo1.txt archivo222.txt archivo2.txt archivo333.txt archivo3.txt archivo.txt
ds-2024@ds2024-VirtualBox:~/Documentos/prueba1$ rm *
ds-2024@ds2024-VirtualBox:~/Documentos/prueba1$ ls
ds-2024@ds2024-VirtualBox:~/Documentos/prueba1$
```

Para ‘ver’ el contenido de un archivo se puede usar el comando **cat**.
La sintaxis es: **cat nombre_archivo.algo**

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cat contenido.txt
Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma, es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario instalar el sistema operativo requerido por el softwareds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Para numerar las líneas podemos escribir: **cat -n contenido.txt**

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cat -n contenido.txt
1 Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas
2 se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma
3 es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el
4 que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario
5 instalar el sistema operativo requerido por el softwareds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Ahora, vamos a visualizar el archivo contenido.txt usando el comando: cat contenido.txt

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cat contenido.txt
Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma, es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario instalar el sistema operativo requerido por el softwareds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Para numerar las líneas podemos escribir: **cat -n contenido.txt**

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cat -n contenido.txt
1 Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas
2 se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma
3 es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el
4 que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario
5 instalar el sistema operativo requerido por el softwareds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Hay varias formas de utilizar el comando cat:

cat > filen.txt : crea un nuevo archivo.

cat archivo1.txt archivo2.txt > archivo3.txt :

fusiona el archivo1.txt con el archivo2.txt y almacena el resultado en el archivo3.txt.

tac archivo.txt: muestra el contenido en orden inverso.

cat > filen.txt: crea un nuevo archivo y se puede escribir sobre él.



ds-2024@ds2024-Virtual

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cat > nuevo.txt
```

```
Esto es un nuevo archivo txt  
para probar el editor cat, una vez que has terminado  
de escribir, presiona CTRL + D para salir del editor
```

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Vamos a visualizar el archivo nuevo.txt:



ds-2024@ds2024-Virtual

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cat nuevo.txt
```

Esto es un nuevo archivo txt
para probar el editor cat, una vez que has terminado
de escribir, presiona CTRL + D para salir del editor

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

cat nuevo.txt contenido.txt > nuevo2.txt :

Combina el contenido de los archivos nuevo.txt y contenido.txt y los guarda en un nuevo fichero(archivo) llamado nuevo2.txt

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cat nuevo.txt contenido.txt > nuevo2.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cat nuevo2.txt
Esto es un nuevo archivo txt.
para probar el editor cat, una vez que has terminado
de escribir, presiona CTRL + D para salir del editorUna de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que
se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma,
es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el
que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario
instalar el sistema operativo requerido por el softwareds-2024@ds2024-VirtualBox:~/Documentos/practica$ 
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ 
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ 
```

tac nuevo.txt :

```
ds-2024@ds2024-VirtualBox: ~/Documentos/practica$ tac nuevo.txt
de escribir, presiona CTRL + D para salir del editorpara probar el editor cat, una vez que has terminado
Esto es un nuevo archivo txt
ds-2024@ds2024-VirtualBox: ~/Documentos/practica$
```

Comando: file

Sintaxis: **file nombrearchivo.txt**

El comando file te permite comprobar un tipo de archivo, ya sea texto, imagen o binario.



```
ds-2024@ds2024-VirtualBox:~$ cd Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ cd practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
contenido1.txt  contenido2.txt  contenido.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ file contenido.txt
contenido.txt: Unicode text, UTF-8 text, with CRLF line terminators
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Comandos zip y unzip

El comando zip te permite comprimir elementos en un archivo ZIP con la relación de compresión óptima. Sintaxis:
zip [opciones] archivozip archivo1 archivo2

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
contenido1.txt  contenido2.txt  contenido.txt  nuevo2.txt  nuevo.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ zip comprimido nuevo.txt nuevo2.txt
adding: nuevo.txt (deflated 21%)
adding: nuevo2.txt (deflated 44%)
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip  contenido1.txt  contenido2.txt  contenido.txt  nuevo2.txt  nuevo.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Utiliza el comando unzip para extraer el archivo comprimido.

Sintaxis: **unzip [opción] nombre_archivo.zip**

unzip comprimido.zip

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ unzip comprimido.zip
Archive: comprimido.zip
replace nuevo.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: nuevo.txt
replace nuevo2.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: nuevo2.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ 
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Comando tar

El comando tar archiva varios elementos en un archivo TAR, un formato similar al ZIP con compresión opcional.

Sintaxis: **tar [opciones] [fichero_archivo] [archivo de destino o directorio]**

tar -cvzf comprimido3.tar /home/ds-2024/Documentos/practica



ds-2024@ds2024-VirtualBox: ~/Documentos/practica\$

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ # vamos a crear una carpeta llamada prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ mkdir prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo.txt prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ pwd
/home/ds-2024/Documentos/practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ # me voy al directorio donde quiero dejar mi archivo tar
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cd prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica/prueba2$ tar -cvzf comprimido3.tar /home/ds-2024/Documentos/practica
tar: Eliminando la '/' inicial de los nombres
/home/ds-2024/Documentos/practica/
/home/ds-2024/Documentos/practica/prueba2/
/home/ds-2024/Documentos/practica/prueba2/comprimido3.tar
/home/ds-2024/Documentos/practica/nuevo.txt
/home/ds-2024/Documentos/practica/comprimido.zip
/home/ds-2024/Documentos/practica/contenido.txt
/home/ds-2024/Documentos/practica/contenido2.txt
/home/ds-2024/Documentos/practica/nuevo2.txt
/home/ds-2024/Documentos/practica/contenido1.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica/prueba2$ ls
comprimido3.tar
ds-2024@ds2024-VirtualBox:~/Documentos/practica/prueba2$
```

```
tar -cvzf comprimido3.tar /home/ds-2024/Documentos/practica
```

- c: Crea un nuevo archivo tar.
- v: Muestra información detallada sobre el proceso de creación del archivo tar.
- z: Comprime el archivo tar usando el algoritmo gzip.
- f: Especifica el nombre del archivo tar.

Otro ejemplo

tar -cvf nuevo_fichero.tar fichero1 directorio2 fichero3 :

Crea “nuevo_fichero.tar”, que contendrá “fichero1”, todo el “directorio2” (y sus subdirectorios) y “fichero3”.

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo.txt prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls prueba2
comprimido3.tar
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ tar cvf nuevo_comprimido.tar nuevo.txt /home/ds-2024/Documentos/practica/prueba2 contenido.txt
tar cvf: orden no encontrada
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ tar -cvf nuevo_comprimido.tar nuevo.txt /home/ds-2024/Documentos/practica/prueba2 contenido.txt
nuevo.txt
tar: Eliminando la '/' inicial de los nombres
/home/ds-2024/Documentos/practica/prueba2/
/home/ds-2024/Documentos/practica/prueba2/comprimido3.tar
tar: Eliminando la '/' inicial de los objetivos de los enlaces
contenido.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo_comprimido.tar nuevo.txt prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ █
```

tar -xvf fichero.tar

Extraerá el contenido de “fichero.tar” y lo dejará en el directorio en el que esté.

```
contenido.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo_comprimido.tar nuevo.txt prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ tar -xvf nuevo_comprimido.tar
nuevo.txt
home/ds-2024/Documentos/practica/prueba2/
home/ds-2024/Documentos/practica/prueba2/comprimido3.tar
contenido.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt home nuevo2.txt nuevo_comprimido.tar nuevo.txt prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls home
```

Comando gzip : gzip fichero1

Comprime los ficheros que recibe como parámetros, eliminando los originales.

gzip fichero1 creará en el mismo directorio la versión comprimida de “fichero1” (“fichero1.gz”) y eliminará “fichero1”



ds-2024@ds2024-VirtualB

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo.txt prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ gzip nuevo2.txt nuevo.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt.gz nuevo.txt.gz prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Comando gunzip

Realiza la operación inversa sobre los archivos comprimidos que recibe como parámetro, restaurando los originales.

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls  
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt.gz nuevo.txt.gz prueba2  
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ gunzip nuevo2.txt  
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls  
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo.txt.gz prueba2  
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ gunzip nuevo.txt  
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls  
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo.txt prueba2  
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Comando head

El comando `head` se utiliza para mostrar las primeras líneas de un archivo de texto. A continuación, se enumeran las opciones más comunes del comando `head`:

`-n`: Especifica el número de líneas que se deben mostrar.

Ejemplo: head -n 3 contenido.txt

Muestra las 3 primeras líneas del archivo contenido.txt

```
ds-2024@ds2024-VirtualBox: ~/Documentos/practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip  contenido1.txt  contenido2.txt  contenido.txt  nuevo2.txt  nuevo_comprimido.tar  nuevo.txt  pruebaz
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ head -n 3 contenido.txt
Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma, es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

-c: Especifica el número de bytes que se deben mostrar.

head -c 10 contenido.txt

Este comando mostrará los primeros 10 bytes del archivo "contenido.txt".

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ head -c 10 contenido.txt  
Una de lasds-2024@ds2024-VirtualBox:~/Documentos/practica$  
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Comando tail

El comando tail muestra las diez últimas líneas de un archivo, lo que resulta útil para comprobar nuevos datos y errores.

Sintaxis: **tail [opción] archivo.txt**

tail -2 contenido.txt

```
ds-2024@ds2024-VirtualBox: ~/Documentos/practica
```

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ tail -n contenido.txt
tail: el número de lineas no es válido: «contenido.txt»
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ tail -2 contenido.txt
que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario
instalar el sistema operativo requerido por el softwareds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

ds-2024@ds2024-VirtualBox:~/Documentos/practica\$

ds-2024@ds2024-VirtualBox:~/Documentos/practica\$ █

Comando diff

Compara el contenido de dos archivos y muestra las diferencias. Se utiliza para alterar un programa sin modificar el código.

diff [opción] archivo1 archivo2

A continuación, se indican algunas opciones aceptables:

- c: muestra la diferencia entre dos archivos en un formulario contextual.
- u: muestra la salida sin información redundante.
- i: hace que el comando diff no distinga entre mayúsculas y minúsculas.

diff contenido.txt nuevo2.txt

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip  contenido1.txt  contenido2.txt  contenido.txt  nuevo2.txt  nuevo_comprimido.tar  nuevo.txt  prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ diff contenido.txt nuevo2.txt
1c1,3
< Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que
---
> Esto es un nuevo archivo txt
> para probar el editor cat, una vez que has terminado
> de escribir, presiona CTRL + D para salir del editorUna de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ █
```

Otro ejemplo con -c:

diff -c contenido.txt nuevo2.txt

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ diff -c contenido.txt nuevo2.txt
*** contenido.txt      2024-04-03 20:17:46.000000000 +0200
--- nuevo2.txt 2024-04-04 09:22:02.000000000 +0200
*****
*** 1,4 ****
! Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que
se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma,
es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el
que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario
--- 1,6 ----
! Esto es un nuevo archivo txt
! para probar el editor cat, una vez que has terminado
! de escribir, presiona CTRL + D para salir del editorUna de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que
se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma,
es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el
que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Con -u:

diff -u contenido.txt nuevo2.txt

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ diff -u contenido.txt nuevo2.txt
--- contenido.txt      2024-04-03 20:17:46.000000000 +0200
+++ nuevo2.txt    2024-04-04 09:22:02.000000000 +0200
@@ -1,4 +1,6 @@
-Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que
+Esto es un nuevo archivo txt
+para probar el editor cat, una vez que has terminado
+de escribir, presiona CTRL + D para salir del editorUna de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que
se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma,
es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto genera problemas si el sistema operativo sobre el
que se va a trabajar no coincide con el que precisa el software que se quiere utilizar, por lo que en estos casos sería necesario
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ █
```

Comando find

Se usa para buscar archivos dentro de un directorio concreto.

find [opción] [ruta] [expresión]

Ejemplo:

find /home -name comprimido3.tar

```
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
comprimido.zip contenido1.txt contenido2.txt contenido.txt nuevo2.txt nuevo_comprimido.tar nuevo.txt prueba2
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls prueba2
comprimido3.tar
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ find /home -name comprimido3.tar
/home/ds-2024/Documentos/practica/prueba2/comprimido3.tar
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ find /home -name nuevo.txt
/home/ds-2024/Documentos/practica/nuevo.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

Actividad 6. Investigar y desarrollar algunos ejemplos con los comandos (en Linux):

- kill
- apt-get
- sudo
- su
- df
- du
- ping
- top
- htop
- ps
- uname
- hostname
- time
- shutdown
- wget
- netstat
- history
- man
- echo
- cal
- ln

Realizar la entrega en un documento .docx y también incluye el fichero .txt con el histórico de los comandos utilizados. Se debe hacer al menos un ejemplo de cada comando.

Edición de archivos: nano, vim





