Modulo 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 icono 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 icono 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

* <° 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…)

* <° Compartidos: Contiene archivos que se pueden encontrar en un ordenador y utilizarse en otro, o incluso compartirse entre usuarios.

Estructura

* <° 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.

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.

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).

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.

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).

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.

/lib/	Librerias esenciales compartidas y módulos de Kernel

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.

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.

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.

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 guardan 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.

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.

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

Seguimos con las características generales: 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.

Seguimos con las 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

Seguimos con las características generales: 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.

Seguimos con las características generales: 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.

Seguimos con las características generales: 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 `ls --color -S -r --exclude='.*'`

En este caso, se utilizan los siguientes modificadores:

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

-S: Ordena los archivos por tamaño.

-r: Ordena los archivos en orden inverso.

--exclude='.*' : Excluye los archivos que comienzan con un punto ("."), que son archivos ocultos.

Actividad 4 (No). Importancia de Linux y Shell para la Ciencia de Datos.

Preparar una presentación donde se explique la importancia y las ventajas que tiene el uso de Linux y la terminal (shell) de Linux para la ciencia de datos. Mencione ejemplos y un caso de uso.

Cantidad **mínima** de diapositivas: **20**

Piense que la presentación es para un público general sin nociones de ciencia de datos.

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.

Vuelve a escribir cd ..

cd ~ : Me lleva al directorio home
cd / : Me lleva al directorio raíz

Para saber mi ubicación actual : **pwd**
El comando **pwd** imprime la ruta del directorio actual
pwd es la abreviación de: print working directory

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.

Otro ejemplo:

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:

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

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

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

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

ls subdir muestra el contenido del subdirectorio “subdir”.

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

Otro ejemplo con: ls –l linux

Actividades Terminal 3 de abr 08:54

ds-2024@ds2024-VirtualBox: /lib

```
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 lib
ds-2024@ds2024-VirtualBox:/lib$ ls
apg                 debug                     gold-ld                                locale              os-probes           rsyslog                    ubuntu-advantage
apparmor            dpkg                      groff                                  lp_solve            os-release          sasl2                      ubuntu-release-upgrader
apt                 emacsen-common            grub                                   lsb                 pam.d               shim                       udev
aspell              environment.d             grub-legacy                            man-db              pcmciautils         snapd                      udisks2
bfd-plugins         evolution-data-server     gvfs                                   memtest86+          pkgconfig           software-properties        ufw
binfmt.d            file                      hdparm                                 mime                pm-utils            speech-dispatcher-modules  unity-settings-daemon-1.0
bluetooth           firewalld                 init                                   modprobe.d          policykit-1         ssl                        update-notifier
brltty              firmware                  initramfs-tools                        modules             polkit-1            sysctl.d                   usrmerge
cnf-update-db       gcc                       ispell                                 modules-load.d      pppd                systemd                    valgrind
command-not-found   girepository-1.0         kernel                                 netplan             pulse-15.99.1+dfsg1 sysusers.d                 X11
compat-ld           gnome-session             klibc                                  networkd-dispatcher python2.7           tc                         x86_64-linux-gnu
console-setup       gnome-settings-daemon-3.0 klibc-K8e6DOmVI9JpyGMLR7qNe5iZeBk.so   NetworkManager      python3             terminfo                   xorg
cpp                 gnome-settings-daemon-42  libreoffice                            nvidia              python3.10          thunderbird                xserver-xorg-video-intel
crda                gnome-shell               linux                                  openssh             python3.11          thunderbird-addons
cups                gnupg                     linux-boot-probes                      openvpn             recovery-mode       tmpfiles.d
dbus-1.0            gnupg2                    linux-sound-base                       os-prober           rhythmbox           ubiquity
ds-2024@ds2024-VirtualBox:/lib$ ls -l linux
total 4
drwxr-xr-x 2 root root 4096 mar 23 10:51 triggers
ds-2024@ds2024-VirtualBox:/lib$
```

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.

```
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.

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

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

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

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

```
ades        Terminal

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

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.**

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:

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$
```

Entramos al subdirectorio y creamos un archivo llamado **test.txt**

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$
```

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:

ades Terminal

```
ds-2024@ds2024-VirtualBox:~/Documentos/carpeta_1$ cd ..
ds-2024@ds2024-VirtualBox:~/Documentos$ mkdir carpeta_2
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_1  carpeta_2  ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ mkdir carpeta_3
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_1  carpeta_2  carpeta_3  ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ ls -l
total 12
drwxrwxr-x 2 ds-2024 ds-2024 4096 abr  3 09:15 carpeta_1
drwxrwxr-x 2 ds-2024 ds-2024 4096 abr  3 09:23 carpeta_2
drwxrwxr-x 2 ds-2024 ds-2024 4096 abr  3 09:23 carpeta_3
-rw-rw-r-- 1 ds-2024 ds-2024    0 abr  3 09:06 ejemplo_2.docx
-rw-rw-r-- 1 ds-2024 ds-2024    0 abr  3 09:07 ejemplo_3.odt
-rw-rw-r-- 1 ds-2024 ds-2024    0 abr  3 09:05 ejemplo.txt
ds-2024@ds2024-VirtualBox:~/Documentos$ rmdir carpeta_3
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
carpeta_1  carpeta_2  ejemplo_2.docx  ejemplo_3.odt  ejemplo.txt
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$
```

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

```
ades        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:

```
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:

```
ades        Terminal

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**

```
ades    Terminal
                                                                ds-2024@ds202
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 exista
  -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 sobreescribir
  -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 exista
                                 (tiene prioridad sobre una opción -i anterior)
  -P, --no-dereference         nunca sigue los enlaces simbólicos en ORIGEN
  -p                            igual que --preserve=mode,ownership,timestamps
      --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                           establece el contexto de seguridad SELinux del fichero de
                                 destino al tipo predeterminado
      --context[=CTX]          como -Z, o 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 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-V
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:

```
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**

```
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***

ades Terminal 3 de abr 11:14

ds-2024@ds2024-VirtualBox: ~/Documentos

```
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**

```
ades    Terminal                                                    3 de abr  11:33
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 -i *.txt
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 *

```
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$
```

Actividad 5. Sistemas de ficheros de Windows y MacOs

- Para los usuarios de Windows , investigar la estructura o sistema de directorios de Windows (la forma como están estructuradas las carpetas y su jerarquía). Explicar sus principales ramas. Replicar los comandos dados en clase (aquellos que te permita la terminal de Windows). Realizar la entrega en un documento de PowerPoint; todos los comandos usados deben mostrarse en el archivo entregable mediante capturas de pantalla. La terminal de Windows que vas a utilizar se llama: **Powershell** y ya viene instalada por defecto. Para ejecutarlo escribe en el buscador de Windows: Powershell y presionas Enter.

- Para los usuarios de Mac, investigar la estructura o sistema de directorios de MacOs (la forma como están estructuradas las carpetas y su jerarquía). Explicar sus principales ramas. Replicar los comandos dados en clase (aquellos que te permita la terminal de Mac). Realizar la entrega en un documento de PowerPoint; todos los comandos usados deben mostrarse en el archivo entregable mediante capturas de pantalla.

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

Antes de usar el comando **cat** vamos a guardar los archivos **contenido.txt**, **contenido1.txt** y **contenido2.txt** en nuestra carpeta compartida. Luego, vamos a copiar nuestro archivo **contenido.txt** que se encuentra en la carpeta compartida dentro de una carpeta que llamaremos **práctica** que esté, a su vez, dentro de Documentos:

Los pasos a continuación:

```
dades    Terminal                                                    3 de abr 20
                                                  ds-2024@ds2024-VirtualBox: ~
ds-2024@ds2024-VirtualBox:~$ cd Documentos
ds-2024@ds2024-VirtualBox:~/Documentos$ mkdir practica
ds-2024@ds2024-VirtualBox:~/Documentos$ ls
practica
ds-2024@ds2024-VirtualBox:~/Documentos$ cd practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ pwd
/home/ds-2024/Documentos/practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cd ..
ds-2024@ds2024-VirtualBox:~/Documentos$ cd ..
ds-2024@ds2024-VirtualBox:~$ cd Escritorio
ds-2024@ds2024-VirtualBox:~/Escritorio$ cd windows
ds-2024@ds2024-VirtualBox:~/Escritorio/windows$ ls
carpeta  contenido.txt  notas_1.txt  notas.txt  prueba_1.txt  prueba.txt
ds-2024@ds2024-VirtualBox:~/Escritorio/windows$ cp contenido.txt /home/ds-2024/Documentos/practica
ds-2024@ds2024-VirtualBox:~/Escritorio/windows$ cd ~
ds-2024@ds2024-VirtualBox:~$ cd Documentos/practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
contenido.txt
ds-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 necesari
     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.

Vamos a copiar los archivos contenido1.txt y contenido2.txt dentro de la carpeta practica:

```
dades    Terminal                                                    3 de abr  21:13
ds-2024@ds2024-VirtualBox: ~/Documentos/practica
ds-2024@ds2024-VirtualBox:~$ cd Documentos/practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
contenido.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ pwd
/home/ds-2024/Documentos/practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ cd ~
ds-2024@ds2024-VirtualBox:~$ cd Escritorio/windows
ds-2024@ds2024-VirtualBox:~/Escritorio/windows$ ls
carpeta  contenido1.txt  contenido2.txt  contenido.txt  jueves.odt  notas_1.txt  notas.txt  prueba_1.txt  prueba.txt
ds-2024@ds2024-VirtualBox:~/Escritorio/windows$ cp contenido1.txt /home/ds-2024/Documentos/practica
ds-2024@ds2024-VirtualBox:~/Escritorio/windows$ cp contenido2.txt /home/ds-2024/Documentos/practica
ds-2024@ds2024-VirtualBox:~/Escritorio/windows$ ls
carpeta  contenido1.txt  contenido2.txt  contenido.txt  jueves.odt  notas_1.txt  notas.txt  prueba_1.txt  prueba.txt
ds-2024@ds2024-VirtualBox:~/Escritorio/windows$ cd ~
ds-2024@ds2024-VirtualBox:~$ cd Documentos/practica
ds-2024@ds2024-VirtualBox:~/Documentos/practica$ ls
contenido1.txt  contenido2.txt  contenido.txt
ds-2024@ds2024-VirtualBox:~/Documentos/practica$
```

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

```
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 editords-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-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 editords-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
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: ~/Do

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/pract
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.

```
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  prueba2
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 líneas 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
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 word o pdf. Se debe hacer al menos un ejemplo de cada comando.

Edición de archivos: nano, vim