TERMINAL Y LÍNEA DE COMANDOS

Comandos básicos de terminal para principiantes

• Ver el nombre de usuario:
    whoami

• Ver el directorio actual:
    pwd (print working directory)

• Listar contenido:
    ls

• Listar los elementos de un directorio en el directorio actual, desde este
    ls ./directorio

• Listar con detalles adicionales como permisos y tamaño:
    ls -l o ll

• Listar incluyendo ocultos:
    ls -a

• Combinar ambas opciones para ver información más útil:
    ls -la

• Incluso en formato legible para humanos:
    ls -lah

• Limpiar terminal:
    Ctrl + L o clear

• Imprimir mensajes en consola (útil en automatizaciones y scripts):
    echo "mensaje"

• Visualizar información del sistema operativo:
    uname -a

• Ver fecha:
    date

• Para acceder al manual de cualquier comando, por ejemplo, para aprender más sobre el uso del comando echo:

man echo

• Colores y símbolos de la terminal:

- Los directorios aparecen en azul y comienzan con la letra "d".
- Mientras que los archivos suelen ser blancos.
- Los caracteres adicionales, letras y números, indican permisos específicos.

Navegación entre directorios en linux con comandos de terminal

• Ir a la ruta absoluta de Linux:

NOTA: Aloja carpetas importantes del sistema como bin, dev, lib64, root y home.

cd /

• Indicar el directorio actual:

NOTA: Se usa en combinación de comandos para referirse al directorio actual.

.

• Retroceder un directorio:

cd ..

• Ir al home del usuario sin importar en que directorio se esté:

cd ~ (Alt + 126)

• Guardar la ruta de la ubicación actual:

pushd

• Recuperar la ruta guardada e ir a ella:

popd

Comandos Linux para crear, mover, copiar y eliminar archivos

• Crear archivo:

touch archivo.formato

• Crear directorio:

mkdir nombre_de_la_carpeta

• Crear estructura de carpetas:
NOTA: -p es para crear los directorios de forma recursiva, sin error si ya existen.

```
mkdir -p carpeta_1/carpeta_2/carpeta_3
```

Resultado:

```
carpeta_1/
└── carpeta_2/
    └── carpeta_3/
```

• Mover archivo o directorio:

```
mv origen destino
```

Ejemplo: Mover la carpeta matematicas al directorio actual:

```
mv escuela/matematicas .
```

NOTA 1: . indica el directorio actual.
NOTA 2: Si el origen es una carpeta se indica la ruta incluyendo la carpeta, si es un archivo también se indica la ruta incluyendo el archivo y su formato.
NOTA 3: Suponiendo que se hace desde la carpeta documentos con una estructura así:

```
documentos/
├── escuela/
│   └── matematicas/
```

Después del comando quedaría así:

```
documentos/
├── escuela/
├── matematicas/
```

• Renombrar archivo o directorio:

```
mv nombre_actual nombre_nuevo
```

• Copiar archivo:

```
cp archivo.formato archivo.formato
```

• Copiar directorio recursivamente, o sea, incluidos los archivos y directorios que contenga:

```
cp -r carpeta carpeta_respaldo
```

• Eliminar archivo:
NOTA: rm se debe usar con PRECAUCIÓN, ya que NO SE PUEDE DESHACER porque no hay papelera en la terminal, es el equivalente a usar Shift + Supr, se recomienda usar rm con -i para que pregunte antes de borrar el archivo.

```
rm archivo.formato
rm -i archivo.formato (Recomendado)
```

• Eliminar directorio recursivamente, o sea, incluidos los archivos y directorios que contenga:
NOTA 1: Al igual que rm para los archivos, también se debe usar con PRECAUCIÓN, ya que NO SE PUEDE DESHACER, se recomienda usar rm -r con i para que pregunte antes de borrar la carpeta.
NOTA 2: -r es para eliminar de forma recursiva y -f para forzarlo sin preguntar.

rm -r carpeta (**CUIDADO**)
rm -ri carpeta (**Recomendado**)
rm -rf carpeta (**NO RECOMENDADO**)

Comandos para explorar y manipular archivos de texto en terminal

• Ver el contenido de un archivo (generalmente .txt):
cat archivo.formato

• Para archivos muy grandes como los CSV se usa:
NOTA: Permite ver el contenido de forma más controlada y con la posibilidad de navegar en él, para salir se presiona la letra q.
less archivo.formato

• Ver las primeras líneas que se indiquen de un archivo:
head -n 10 archivo.formato

• Ver las ultima líneas que se indiquen de un archivo:
tail -n 10 archivo.formato

• Numerar las líneas de un archivo:
nl archivo.formato

• Saber la cantidad de líneas, palabras y caracteres (se muestran en ese orden):
wc archivo.formato

• Saber la cantidad de líneas:
wc -l archivo.formato

• Saber la cantidad de palabras:
wc -w archivo.formato

• Saber la cantidad caracteres:
wc -c archivo.formato

• Imprimir columnas de un archivo CSV:

NOTA: Los separadores se indican con -F y las columnas específicas con $.

awk '{print $1}' archivo.formato (Imprime la primera linea)

awk -F"," '{print $1, $3}' archivo.csv (Imprime las columnas 1 y 3)

Uso de wildcards (comodines) para búsquedas masivas en la terminal

• Listar archivos por extensión:

ls *.txt

• Listar archivos que empiecen por cierta palabra, por ejemplo "file":

ls file*

• Listar archivos que coincidan con un carácter específico:

NOTA: ? lista los archivos con cualquiera que sea el carácter en la posición que se ponga.

ls file?.txt

• Listar coincidencias específicas con corchetes y asteriscos, por ejemplo, que terminen con o antes del punto:

ls *[o].*

• Listar por patrones, por ejemplo, por el formato de archivos:

ls *.{formato1,formato1}

NOTA 1: Este comando marca error si no hay coincidencias con uno de los patrones (o si solo se checa un patrón), para evitarlo (al menos en zsh) se usa:

setopt +o nomatch

ls *.{formato1,formato1}

setopt nomatch

NOTA 2: Para saber que Shell se esta usando es con echo $SHELL.

• Mover archivos con el wildcard *, por ejemplo, mover todos los archivos .txt a la carpeta backup:

mv *.txt backup/

Comandos GREP y FIND para búsquedas avanzadas en Linux

• Listar las líneas que coincidan con una búsqueda de texto en un archivo:

NOTA: -i es para no diferenciar entre minúsculas y mayúsculas.

grep -i "texto" archivo.formato

• Contar las líneas que coincidan en un archivo:
grep -ic "texto" archivo.formato

• Listar las líneas que **NO** coincidan en un archivo:
grep -iv "texto" archivo.formato

• Buscar carpetas:
find . -type d -name "*"

NOTA: El primer parámetro, ".", también puede ser "/", porque se refiere a partir de donde hará la búsqueda. -type es para d (directorio) o f (file). Y -name es para "*", que también puede ser un nombre especifico. Este comando significa: Sin importar el nombre, busca todas las carpetas (incluidas las ocultas) a partir de la ubicación actual (por el .).

• Buscar archivos grandes:
find . -type f -size +1M

Tipos de comandos en Linux y cómo identificarlos

• Saber si un comando es alias:
type comando

• Ver ruta del comando:
which comando

• Ver todas las rutas relacionadas del comando:
whereis comando

• Ver descripción breve:
whatis comando

Redirecciones de terminal en Linux con operadores básicos

• Enviar la salida de un comando a un archivo (el archivo se crea en ese momento):
NOTA: Si se vuelve a hacer el mismo comando, pero con otro texto, el texto más reciente sustituirá TODO LO QUE CONTENGA EL ARCHIVO, para que no pase eso e irlo agregando al mismo se usa con doble >>. Esto es igual al agregar errores a un archivo.
echo "texto" > archivo.formato

• Agregar contenido a archivo existente (se agrega al final del archivo):
    echo "texto" >> archivo.formato

• Encadenar la salida de un comando con la entrada de otro con | (Alt + 124):
    echo "hola" | lolcat

• Capturar errores en un archivo (el archivo se crea en ese momento):
NOTA 1: Si se vuelve a hacer el mismo comando, pero con otro error, el error más reciente sustituirá TODO LO QUE CONTENGA EL ARCHIVO, para que no pase eso e irlo agregando al mismo se usa con doble >>.
NOTA 2: El flujo de errores estándar (*standard error*) se referencia con el número 2.
    ls archivo_inexistente 2> errores.log

• Agregar varios errores a un archivo existente (se agrega al final del archivo):
    ls archivo_inexistente 2>> errores.log

• Guardar salida y errores en un solo archivo:
    comando &> output.log

Operadores de control para encadenar comandos en Linux

• Encadenar sin condición, o sea, que los comandos se ejecuten uno detrás del otro sin importar si alguno falla con ; (operador secuencial):
    comando1; comando2

• Ejecutar solo si el anterior fue exitoso con && (y):
    comando1 && comando2

• Ejecutar, aunque el anterior falle con || (o):
    comando1 || comando2

• Combinar condiciones múltiples:
    comando1 && comando2 || comando3

Configuración de alias permanentes en terminal Linux

• Ver alias actuales:
    alias

• Crear alias temporal:

NOTA: El comando a ejecutar va entre comillas simples.

alias nombre_alias='comando a ejecutar'

• Crear alias permanente con >>, agregándolo al archivo de configuración. Para saber cuál es, se ejecuta; **echo $SHELL**, si es bash, será .bashrc, si es zsh, será .zshrc:

NOTA: Además de que el comando a ejecutar va entre comillas simples, el nombre del alias hasta el comando va entre comillas dobles.

echo alias “nombre_alias='comando a ejecutar'” >> ~/.bashrc

- Para que los cambios sean efectivos inmediatamente, se recarga el archivo con:

source ~/.bashrc

Gestión de permisos en archivos y directorios de Linux

• Los permisos son las letras de la izquierda que salen al listar los archivos. Después de la d o - del inicio, cada tres letras o guiones se refiere a los permisos de usuario, grupos y otros (en ese orden). A excepción del carácter inicial, si entre las letras hay un guion en lugar de una de letra, significa que el archivo no tiene ese permiso para ese usuario, grupo u otro caso.

• Significado de las letras:

- d es para directorio
- Guion (-) es para archivo
- r = read = valor 4
- w = write = valor 2
- x = execute = valor 1

• Ver permisos:

ll -a (Equivalente a ls -lah)

• Asignar permisos con letras:

chmod u+x archivo.formato (Da permiso de ejecución solo al usuario)

• Al asignar permisos con números, cada número equivale a la suma de cada uno de los permisos por usuario, grupo u otros:

- chmod 755 archivo.formato (Da todos los permisos al usuario, y para grupo y otros solo de lectura y ejecución)
- chmod 644 archivo.formato (Da permisos de lectura y escritura al usuario, y para grupo u otros solo de lectura)

NOTA: 755 es una buena configuración para scripts compartidos y 744 la más segura.

• Aplicar permisos de forma recursiva a una carpeta, o sea, también a todo lo que contenga:

```
chmod -R 755 carpeta/
```

NOTA: Aplicar cambios recursivamente a un directorio y sus contenidos debe ser con cautela ya que:

- **Aplicarlo a todos los archivos podría generar conflictos.**
- **Es preferible limitar sus efectos usando patrones específicos mediante herramientas como find .**
- **Linux suele proteger estas operaciones recursivas por seguridad, evitando que afecten negativamente otros archivos o scripts.**

Variables de entorno en Linux: creación y configuración global

• Una variable de entorno es un tipo especial de variable que guarda información que podrá ser utilizada por el sistema y diversas aplicaciones. Se invocan en la shell utilizando el símbolo de dólar ($) seguido por el nombre de la variable.

• Ver variables actuales:

```
env | sort | less
```

• Crear variable temporal (solo para la sesión actual de terminal):

```
my_var="valor"
```

• Verificarla:

```
echo $my_var
```

• Para que esté disponible en múltiples sesiones de terminal hay que exportarla:
NOTA: Esto aun no la hace permanente, ya que si se cierran todas las terminales desaparecerá.

```
export my_var="valor"
```

• Hacerla permanente en zsh:

```
echo 'export my_var="valor"' >> ~/.zshrc
source ~/.zshrc (Recargar la configuración)
cat ~/.zshrc (Checar si la variable se agregó al final del archivo)
```

• Hacerla permanente en bash:
    echo 'export my_var="valor"' >> ~/.bashrc
    source ~/.bashrc (Recargar la configuración)
    cat ~/.bashrc (Checar si la variable se agregó al final del archivo)

Uso de APT para gestionar paquetes en Linux

• Actualizar la información de los paquetes instalados:
    sudo apt update

• Revisar actualizaciones disponibles:
    apt list --upgradeable

• Actualizar paquetes instalados:
    sudo apt upgrade

• Instalar paquete:
    sudo apt install nombre_paquete

• Ver información de un paquete:
    apt show nombre_paquete

• Eliminar paquete:
    sudo apt remove nombre_paquete

• Eliminar completamente junto con sus dependencias y archivos de configuración:
    sudo apt purge nombre_paquete

Instalación y uso básico de Homebrew en macOS

• Instalar Homebrew (recomendable usar *iTerm* o la Terminal por defecto con la shell de ZSH):
    /bin/bash -c "$(curl -fsSL
    https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

• Confirmar que se instaló checando la versión:
    brew --version

• Buscar paquete:
    brew search nombre_paquete

• Ver información de un paquete:
    brew info nombre_paquete

• Instalar paquete:
    brew install nombre_paquete

• Listar paquetes instalados:
    brew list

• Actualizaciones generales:
    brew upgrade

• Actualizar brew:
    brew update

• Eliminar paquete y limpiar:
    brew uninstall nombre_paquete
    brew cleanup

Procesos en foreground y background en la terminal

• Enviar proceso a segundo plano:
NOTA: El comando o conjunto de comandos deben ir entre paréntesis.
    (comando) &

• Listar los procesos actuales en background con sus estados (activos o pausados):
    jobs

• Pausar proceso:
    Ctrl + Z

• Detener proceso:
    Ctrl + C

• Traer/reanudar un proceso al foreground:
    fg + %processID

• Traer/reanudar un proceso pausado o suspendido en background:
    bg + % processID

Administración de procesos en Linux con PS, Top y Kill

• Ver una "fotografía" actual de los procesos activos:

```
ps aux
```

• Buscar proceso:

```
ps aux | grep -i nombre_proceso
```

• Monitor en tiempo real:

NOTA: El valor "NI" determina la prioridad de cada proceso, donde valores negativos indican procesos críticos (mayor prioridad y recursos asignados), mientras que valores positivos son menos prioritarios. q para salir.

```
top
```

• Instalar htop:

```
sudo apt install htop
```

• Mejoras de htop sobre top:

- Visualización gráfica y detallada de memoria y CPU.
- Búsqueda eficiente de procesos mediante teclas de función (como F3 para búsqueda por nombre).
- Representación visual en formato de árbol de los procesos padre e hijo usando la tecla F5.

• Identificar el PID (process ID) del proceso a finalizar:

NOTA: El resultado también incluye el propio grep nombre_ proceso (porque el comando también contiene el nombre del proceso a buscar).

```
ps aux | grep nombre_proceso
```

• Finalizar proceso:

```
kill -9 PID
```

Empaquetado y compresión de archivos con TAR y GZIP en Linux

• Empaquetar consiste en combinar múltiples archivos o carpetas en un único paquete que el sistema operativo reconocerá como un solo archivo.

• La compresión reduce significativamente el tamaño de un archivo eliminando datos redundantes. El estándar más común en terminal Linux es gzip.

• Empaquetar con TAR:
    tar -cvf nombre_archivo.tar carpeta_o_archivos

• Comprimir archivo previamente empaquetado:
NOTA: El resultado será un archivo con extensión .tar.gz.
    gzip nombre_archivo.tar

• Empaquetar y comprimir en un solo comando:
    tar -czvf nombre_archivo.tar.gz carpeta_o_archivos

• Descomprimir:
    gunzip nombre_archivo.tar.gz

• Desempaquetar:
    tar -xvf nombre_archivo.tar

• Descomprimir y desempaquetar en un solo comando:
    tar -xzvf nombre_archivo.tar.gz

NOTA: Si un archivo comprimido se extrae dentro de la misma carpeta del archivo original, el extraído sobrescribirá el original sin preguntar, hacer copia de seguridad ni crear el archivo extraído con (1) al final, así que se debe checar el contenido antes de extraerlo, pero lo más seguro es extraerlo en una subcarpeta.

• Ver el contenido de un archivo comprimido antes de extraer:
    tar -tzvf nombre_archivo.tar.gz

• Extraer en una subcarpeta:
    mkdir extraido && tar -xzvf nombre_archivo.tar.gz -C extraido

• Empaquetar y comprimir son técnicas independientes, se puede crear un paquete sin comprimir o comprimir archivos individuales sin empaquetarlos.

Editores de texto Vim y Nano en la terminal

• Abrir archivo en vim:
    vim nombre_archivo.txt

• Escribir en vim (modo inserción)
i

• Salir del modo inserción (también activa el modo comando para ejecutar acciones)

esc

• Guardar en vim:

:w

• Salir vim:

:q

• Guardar y salir de vim:

:wq

• Salir sin guardar de vim:

:q!

• Elimina toda la línea actual:

dd

• Navega al inicio del archivo:

gg

• Navega a una línea exacta:

:numero

• Abrir archivo en nano:

nano nombre_archivo.formato

• Guardar en nano:

Ctrl + o

• Salir de nano:

Ctrl + x

• Cortar líneas seleccionadas en nano:

Ctrl + k

• Pegar texto previamente cortado en nano:

Ctrl + u

• Ver todas las opciones disponibles en nano (desde nano):

Ctrl + g

Tmux para gestionar múltiples terminales y paneles

• Tmux es una herramienta poderosa que facilita el trabajo simultáneo en varias terminales dentro de una sola ventana. Trabaja mediante lo que se denomina "prefijo", una combinación de teclas específica (generalmente Ctrl + B).

• Instalar Tmux en Linux:
    sudo apt install tmux

• Instalar Tmux en Mac:
    brew install tmux

• Dividir verticalmente:
    Ctrl + B luego %

• Dividir horizontalmente:
    Ctrl + B luego "

• Crear nueva ventana (sesion):
    Ctrl + B luego C

• Renombrar ventana actual:
    Ctrl + B luego ,

• Cambiar entre ventanas:
    Ctrl + B seguido del número índice de la ventana (0, 1, 2)

• Moverse entre paneles:
    Ctrl + B luego flechas (Por cada desplazamiento)

• Cerrar panel/ventana (cuando ya no hay más paneles):
    exit o Ctrl + D

• Listar las sesiones de tmux:
    tmux ls

• Volver a la sesion de tmux si la terminal se cerró:
    tmux attach

Comandos de red en la terminal para verificar conectividad

• Ver interfaces de red:
    ip a

• Ver tabla de ruteo:
    ip r

• Probar conexión (Ctrl + Z para detener):
    ping www.google.com

• Ver HTML de una web:
    curl www.google.com

• Guardar el HTML de una web:
    curl www.google.com > index.html

• Descargar archivo:
    wget https://url/archivo.zip

• Escanear puertos de la computadora, determinando servicios expuestos o funcionalidades activas (sudo apt install nmap para instalarlo):
    nmap 192.168.1.1

• Rastrear el recorrido exacto de paquetes enviados hacia un destino, evidenciando qué servidores atraviesa en camino al recurso solicitado (sudo apt install traceroute para instalarlo):
    traceroute www.google.com

Personalización de terminal con ZSH y temas avanzados

• Instalar ZSH:
    sudo apt install zsh

• Instalar Oh My ZSH:
    sh -c "$(curl -fsSL https://raw.github.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"

• Instalar Powerlevel10k:
    git clone https://github.com/romkatv/powerlevel10k.git $ZSH_CUSTOM/themes/powerlevel10k

• Antes de activar Powerlevel10k, instala previamente la fuente Meslo NerdFont para visualizar correctamente iconos y símbolos específicos.

• Activar Powerlevel10k modificando la variable ZSH_THEME en el archivo .zshrc por:

"powerlevel10k/powerlevel10k"
source ~/.zshrc (Recargar la configuración y configurarlo)