



ejreyesh /
documentos



<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings



main

documentos / tareas / 2023-05-12.md



Go to file

t



ejreyesh Create 2023-05-12.md

5 months ago



210 lines (166 loc) · 7.89 KB

Preview

Code

Blame

Raw



Comandos Linux



Los Comandos Linux son palabras reservadas que usa el sistema operativo para ejecutar determinadas acciones usando una terminal o línea de comandos. Una terminal Linux es una interfaz o programa con la que se puede ejecutar a los comandos.

Comandos más utilizados

man

Permite conocer el uso de todos los comandos de Linux, al ejecutarlo se muestra una lista de todos los comandos existentes incluyendo información como nombre, sinopsis, descripción, opciones, estado de salida, valores de devolución, errores, archivos, versiones, ejemplos, etc. Por ejemplo, para conocer el comando `cd` y sus opciones, se tiene que ejecutar el comando `man cd` para conocer su descripción y uso.

```
man [command/tool name]
```



whoami

Indica el usuario que tiene la sesión activa en la computadora.



sudo [↗](#)

Este comando significa **superuser do**, y permite actuar como super usuario o usuario root mientras se realiza la ejecución de un comando específico. Es la forma en que Linux se protege a sí mismo y evita que los usuarios modifiquen accidentalmente el sistema de archivos de la máquina o instalen paquetes inapropiados. Es común utilizar este comando para instalar software o para editar archivos fuera del directorio personal del usuario:

```
sudo apt install gimp  
sudo cd /root/
```



Al intentar ejecutar un comando con esta instrucción, la terminal solicita la contraseña del administrador.

apt [↗](#)

Permite acceder al gestor de paquetes a través de la línea de comandos, es decir, con este comando se podrá instalar, actualizar y eliminar distinto software instalado en los equipos.

```
sudo apt install application_name
```



clear [↗](#)

Limpia la terminal del texto que se vaya geneando.

```
clear
```



history [↗](#)

Indica el historial de comandos Linux que se han ejecutado desde la primera vez que se instaló el S.O.

```
history
```



exit [↗](#)

El comando `exit` se puede utilizar para terminar una sesión de shell y, en la mayoría de los casos, cerrar automáticamente el terminal que se está utilizando.

```
exit
```



ls [↗](#)

Permite listar el contenido del directorio que se desee, incluyendo archivos y otros directorios anidados.

```
ls
```



`ls` tiene muchas opciones, así que se sugiere el uso del flag `--help`. Este flag devuelve todos los flags que pueden utilizarse junto a `ls`.

pwd [↗](#)

Muestra la ruta absoluta del directorio en el que se encuentra actualmente. Para utilizarlo, basta con escribir `pwd` en el terminal:

```
pwd
```



cd [↗](#)

Su nombre significa "cambiar de directorio" y, permite situarse dentro del directorio al que se intenta acceder. Por ejemplo, si se está situado dentro del directorio Documentos y se quiere acceder a una de sus subcarpetas llamada Vídeos, se debe emplear el siguiente comando:

```
cd Videos
```



Otra alternativa es proporcionar la ruta absoluta de la carpeta:

```
cd /home/ubuntu/Documents/Videos
```



Combinaciones del comando `cd` [↗](#)

- Ir a la carpeta de inicio: `cd`
- Situarse en un directorio anterior al actual: `cd ..`

cp [↗](#)

Diseñado para copiar archivos y directorios directamente desde la terminal de Linux. Para utilizar el comando `cp`, basta con escribirlo junto con los nombres de los archivos de origen y destino.

```
cp file_to_copy.txt new_file.txt
```



Para copiar directorios enteros es necesario utilizar la bandera de recursión `-r`.

```
cp -r dir_to_copy/ new_copy_dir/
```



rm [↗](#)

Comando utilizado para eliminar archivos y directorios. Sin embargo, es necesario tener cuidado al emplearlo, porque es difícil (aunque no imposible) recuperar los archivos eliminados de esta manera.

Para borrar un archivo el comando necesario es:

```
rm file_name.txt
```



Si se desea eliminar un directorio vacío se puede emplear la flag recursiva `-r`:

```
rm -r dir_to_remove/
```



Para eliminar un directorio con contenido en tu interior, es necesario utilizar el flag force `-f` y la flag de recursión `-r`:

```
rm -rf dir_with_content_to_remove/
```



cat [↗](#)

Se utiliza, principalmente, para ver el contenido de un archivo o para crear un archivo con permisos de escritura.

Para mostrar el contenido del archivo en la terminal se usa el comando:

```
cat file_name
```



Para crear un archivo con el nombre indicado y al mismo tiempo editarlo (agregando el texto que se desee) se utiliza el comando:

```
cat > nombre-archivo.txt
```



Para copiar el contenido de varios archivos a un tercero:

```
cat archivo1 archivo2 > archivo3.txt
```



mkdir [↗](#)

Se utiliza para crear directorios, la estructura de los comandos se compone de la directiva `mkdir` seguido del nombre del directorio que desee crear. También es posible usar `/` para crear un directorio dentro de otro directorio.

El siguiente comando permite crear un directorio con el nombre establecido.

```
mkdir directory_name
```



El comando mostrado a continuación crea un directorio2 dentro del directorio1.

```
mkdir directorio1/directorio2
```



rmdir [↗](#)

Comando diseñado para eliminar un directorio. El directorio se podrá borrar únicamente si no tiene contenido dentro.

El comando mostrado a continuación elimina al directorio1 siempre y cuando este vacío.

```
rmdir Directorio1
```



El comando siguiente borra el directorio2 que se ubica dentro del directorio1.

```
rmdir Directorio1/Directorio2
```



chmod [↗](#)

El comando `chmod` permite cambiar los permisos un archivo o directorio rápidamente. Existen dos formas de asignar permisos, se denominan notación simbólica y notación octal.

Los permisos básicos que puede tener un archivo son:

- r (leer)
- w (escribir)
- x (ejecutar)

La estructura básica de este tipo de comandos es:

```
chmod <modificadores> <permiso> nombre_archivo
```



Para obtener más información con respecto al uso de este comando dirigirse al siguiente enlace [permisos en Linux](#).

./

Esta instrucción es una notación y permite a la línea de comandos correr un archivo ejecutable mediante el uso de cualquier intérprete instalado en el sistema. La ejecución se realiza directamente desde el terminal.

```
./
```



ps

Permite visualizar los procesos que la sesión de shell actual está ejecutando. Imprime información útil sobre los programas que está ejecutando, como el ID del proceso, la hora y el nombre del comando.

```
ps
```



ll

Enumera los nombres de los archivos y directorios contenidos dentro del directorio actual, además muestra información como los permisos, tamaño, fecha y hora de creación e información del propietario del fichero.

```
ll
```



sudo apt-get update

Actualiza la lista de paquetes disponibles y sus versiones, sin embargo, no instala o actualiza ningún paquete. La información obtenida mediante este comando proviene directamente de los servidores con repositorios definidos dentro del sistema operativo.

```
sudo apt update
```



apt-get upgrade

Generalmente es empleado después de ejecutar el comando `sudo apt update`, ya que instala las nuevas versiones de cada uno de los paquetes instalados dentro del equipo. Dicha actualización se hace respetando la configuración del software previamente instalado.

```
sudo apt upgrade
```



sudo su

Este comando permite pasar a la cuenta root o de super usuario. La cuenta a la que se accede es la cuenta con los permisos más elevados. Este usuario tiene permisos de lectura, escritura y ejecución de cualquier aplicación del sistema (acceso administrativo en general); a diferencia de los usuarios normales que tienen acceso limitado a ciertas tareas por razones de seguridad.

```
sudo su
```



Referencias bibliográficas

Gustavo, B. (2019). Comando cat de Linux – Con ejemplos de uso.

<https://www.hostinger.mx/tutoriales/comando-cat-linux>

Diaz, D. (2021). Los 40 Comandos de Linux Más Utilizados que Debes Conocer.

<https://kinsta.com/es/blog/linux-comandos/>

Montufar, K. (s/f). Comando (sudo su) · Comandos linux "Ubuntu server".

https://kevinmontufar.gitbooks.io/comandos-linux-ubuntu-server/content/comando_sudo_su.html