Tutorial Completo: Conceptos Básicos Esenciales de Linux

Fecha Actual: Jueves, 10 de abril de 2025

Ubicación: Málaga, Andalucía, España

0. Introducción: ¿Qué es Linux y Por Qué Aprender lo Básico?

- **¿Qué es Linux?** Linux no es un sistema operativo completo en sí mismo, sino el **kernel**, el núcleo central que gestiona el hardware del ordenador. Lo que usamos son **distribuciones de Linux** (como Ubuntu, Fedora, Debian, Mint, Manjaro, etc.), que combinan el kernel Linux con una colección de software (entorno de escritorio, utilidades, aplicaciones) para crear un sistema operativo usable. Linux es conocido por ser **Open Source** (código abierto), estable, seguro y muy flexible.
- **La Terminal (o Consola):** Aunque Linux moderno tiene interfaces gráficas amigables, la **terminal** (o línea de comandos) sigue siendo una herramienta increíblemente poderosa y eficiente para interactuar con el sistema. Aprender los comandos básicos te da un control mucho mayor.
- **La Shell:** Es el programa que interpreta los comandos que escribes en la terminal (ejemplos: Bash, Zsh, Fish). Bash es el más común.

¿Cómo abrir una terminal? Busca "Terminal", "Konsole", "xterm" o similar en el menú de tu distribución, o usa un atajo como Ctrl+Alt+T en muchas de ellas. Verás un **prompt**, algo como tu_usuario@nombre_equipo:~$, esperando tus comandos.

1. Navegación Básica por el Sistema de Archivos

Linux organiza todo en una estructura de directorios jerárquica, empezando desde la raíz (/).

- pwd **(Print Working Directory):** Muestra el directorio (carpeta) en el que te encuentras actualmente.

Bash

```
pwd

# Salida de ejemplo: /home/tu_usuario
```

- ls **(List):** Lista el contenido del directorio actual.

Bash

```
ls

# Salida: Documentos  Descargas  Musica  archivo.txt
```

- ls -l **(Long format):** Muestra una lista detallada, incluyendo permisos, propietario, tamaño y fecha (¡lo veremos en detalle!).

Bash

```
ls -l

# Salida de ejemplo:

# drwxr-xr-x 2 tu_usuario tu_grupo 4096 abr 10 18:00 Documentos

# -rw-r--r-- 1 tu_usuario tu_grupo  123 abr  9 10:30 archivo.txt
```

- ls -a **(All):** Muestra todos los archivos, incluidos los ocultos (los que empiezan con un punto .).

Bash

```
ls -a

# Salida: .  ..  .bashrc  Documentos  Descargas  Musica  archivo.txt

# (. representa el directorio actual, .. representa el directorio padre)
```

- Puedes combinar opciones: ls -la o ls -al.

- cd **(Change Directory):** Cambia de directorio.

Bash

```
# Ir al directorio Documentos (si está dentro del actual)

cd Documentos

pwd

# Salida: /home/tu_usuario/Documentos


# Volver al directorio padre (uno arriba)

cd ..

pwd

# Salida: /home/tu_usuario


# Ir directamente a un directorio usando la ruta absoluta (desde la raíz /)

cd /var/log
```

Ir a tu directorio "home" directamente desde cualquier sitio

cd

o

cd ~

Ir al directorio anterior en el que estabas

cd -

2. Manejo de Archivos y Directorios

- mkdir **(Make Directory):** Crea un nuevo directorio.

Bash

mkdir MiNuevaCarpeta

ls

Verás MiNuevaCarpeta listada

    - mkdir -p Proyectos/Web/Frontend**:** Crea toda la estructura de directorios anidados si no existen (-p de parents).

- rmdir **(Remove Directory):** Elimina un directorio **vacío**.

Bash

rmdir MiNuevaCarpeta

    - Si la carpeta no está vacía, dará error. Para eliminar carpetas con contenido, usa rm (ver abajo).

- touch**:** Crea un archivo vacío o actualiza la fecha de modificación de uno existente.

Bash

touch mi_archivo.txt

ls -l mi_archivo.txt

Verás un archivo nuevo con tamaño 0

- cp **(Copy):** Copia archivos o directorios.

Bash

Copiar un archivo en el mismo directorio con otro nombre

cp mi_archivo.txt mi_archivo_copia.txt

Copiar un archivo a otro directorio

cp mi_archivo.txt Documentos/

Copiar un directorio y todo su contenido (recursivamente)

mkdir MiOtraCarpeta

touch MiOtraCarpeta/otro.txt

cp -r MiOtraCarpeta/ CopiasDeSeguridad/

Necesitas la opción -r para directorios

- mv **(Move):** Mueve o renombra archivos y directorios.

Bash

Renombrar un archivo

mv mi_archivo.txt archivo_renombrado.txt

Mover un archivo a otro directorio

mv archivo_renombrado.txt Documentos/

Mover un directorio

mv CopiasDeSeguridad/ /tmp/

- rm **(Remove):** Elimina archivos (¡CON CUIDADO!).

Bash

rm mi_archivo_copia.txt

    - rm -i archivo_peligroso.txt**:** Pide confirmación antes de borrar (-i de interactivo).
    - rm -r directorio_a_borrar/**:** Elimina un directorio y TODO su contenido recursivamente. **¡¡EXTREMADAMENTE PELIGROSO SI TE EQUIVOCAS!! ¡¡NO HAY PAPELERA DE RECICLAJE EN LA TERMINAL POR DEFECTO!!** Úsalo con máxima precaución.

- rm -rf directorio_a_borrar/**:** Igual que -r, pero fuerza el borrado sin pedir confirmación (-f de force). **¡¡DOBLEMENTE PELIGROSO!!** Evítalo si no estás 100% seguro.

3. Visualización de Archivos

- cat **(Concatenate):** Muestra el contenido completo de uno o más archivos en la terminal.

Bash

cat archivo_renombrado.txt

Si el archivo es largo, llenará la pantalla

- less**:** Muestra el contenido de un archivo página por página. Muy útil para archivos largos.

Bash

less /var/log/syslog

- Usa las flechas arriba/abajo, RePág/AvPág para moverte.
- Escribe /palabra_a_buscar y presiona Enter para buscar.
- Presiona q para salir.

- head**:** Muestra las primeras líneas de un archivo (10 por defecto).

Bash

head /etc/passwd

Mostrar las primeras 5 líneas

head -n 5 /etc/passwd

- tail**:** Muestra las últimas líneas de un archivo (10 por defecto).

Bash

tail /var/log/syslog

Mostrar las últimas 20 líneas

tail -n 20 /var/log/syslog

Opción útil: seguir viendo el final del archivo mientras se actualiza (logs)

tail -f /var/log/syslog

(Presiona Ctrl+C para detener el seguimiento)

4. Obtener Ayuda

- man **(Manual):** Muestra la página del manual de un comando. Es la ayuda más completa.

Bash

man ls

man cp

man chmod

    - Navega igual que con less (flechas, / para buscar, q para salir).

- **Opción** --help**:** La mayoría de los comandos ofrecen una ayuda rápida con sus opciones.

Bash

ls --help

mkdir --help

5. Usuarios y Grupos

Linux es un sistema multiusuario. Cada persona que usa el sistema tiene una cuenta de **usuario**. Los usuarios pertenecen a uno o más **grupos**. Los permisos se basan en quién es el propietario (usuario) del archivo/directorio y a qué grupo pertenece.

- whoami**:** Muestra tu nombre de usuario actual.
- id**:** Muestra tu información de usuario y los grupos a los que perteneces.
- groups**:** Muestra solo los grupos a los que perteneces.

6. Permisos de Archivos y Directorios (¡Esencial!)

Esta es una de las partes más importantes y a veces confusas de Linux. Controla quién puede hacer qué con los archivos y directorios.

- **Visualización con** ls -l**:**

Bash

ls -l mi_script.sh

Salida: -rwxr-xr-- 1 juan staff 150 abr 10 18:30 mi_script.sh

Desglose de la primera parte (-rwxr-xr--):

    - **Primer carácter (-): Tipo de archivo.**
        - -: Archivo regular.

        - d: Directorio.
        - l: Enlace simbólico (acceso directo).
        - Otros (menos comunes): c (carácter), b (bloque), s (socket), p (pipe).
    - **Los siguientes 9 caracteres son los permisos, divididos en 3 grupos de 3:**

1. rwx **(Usuario/Propietario):** Permisos para el dueño del archivo (juan en el ejemplo).

2. r-x **(Grupo):** Permisos para los miembros del grupo al que pertenece el archivo (staff en el ejemplo).

3. r-- **(Otros):** Permisos para cualquier otro usuario del sistema.

    - **Significado de** r**,** w**,** x**:**
        - r **(Read - Lectura):**
            - Para archivos: Permite ver el contenido.
            - Para directorios: Permite listar el contenido del directorio (ls).
        - w **(Write - Escritura):**
            - Para archivos: Permite modificar o borrar el archivo.
            - Para directorios: Permite crear, borrar o renombrar archivos *dentro* de ese directorio (¡incluso si no tienes permiso w sobre los archivos mismos!).
        - x **(Execute - Ejecución):**
            - Para archivos: Permite ejecutar el archivo (si es un programa o script).
            - Para directorios: Permite entrar (cd) al directorio. **¡Necesitas** x **para poder hacer** ls **o acceder a archivos dentro, aunque tengas** r**!**

- Modificación con chmod (Change Mode):

Hay dos formas principales de usar chmod:

1. **Modo Simbólico (más intuitivo):** Usa letras para indicar a quién (u=usuario, g=grupo, o=otros, a=todos) y qué hacer (+=añadir permiso, -=quitar permiso, ==establecer permiso exacto).

Bash

Añadir permiso de ejecución al propietario (usuario)

chmod u+x mi_script.sh

Quitar permiso de escritura al grupo y a otros

chmod go-w archivo_sensible.txt

Dar permisos de lectura y escritura al usuario, y solo lectura al grupo y otros

chmod u=rw,go=r mi_documento.txt

Añadir permiso de lectura a todos

chmod a+r reporte.pdf

2. **Modo Octal (más rápido para expertos):** Usa números (base 8) para representar los permisos. Cada permiso tiene un valor: r=4, w=2, x=1. Se suman para cada grupo (Usuario, Grupo, Otros).

- rwx = 4 + 2 + 1 = 7
- rw- = 4 + 2 + 0 = 6
- r-x = 4 + 0 + 1 = 5
- r-- = 4 + 0 + 0 = 4
- -wx = 0 + 2 + 1 = 3
- -w- = 0 + 2 + 0 = 2
- --x = 0 + 0 + 1 = 1
- --- = 0 + 0 + 0 = 0

Bash

Ejemplo: -rwxr-xr-- (dueño rwx, grupo r-x, otros r--)

rwx = 7

r-x = 5

r-- = 4

Comando:

chmod 754 mi_script.sh # (Aunque 754 es raro, 755 es más común para scripts)

Permisos comunes:

Archivos de datos normales: rw-r--r-- (644)

chmod 644 mi_documento.txt

Directorios normales: rwxr-xr-x (755) - ¡Necesitas 'x' para entrar!

chmod 755 MiDirectorio/

Archivos/Scripts ejecutables para el dueño: rwx------ (700) o rwxr-xr-x (755)

chmod 700 mi_script_privado.sh

- **Cambiar Propietario y Grupo:**
    - chown (Change Owner): Cambia el propietario (y opcionalmente el grupo). **Normalmente requiere** sudo.

Bash

Cambiar el dueño a 'nuevo_usuario'

sudo chown nuevo_usuario mi_archivo.txt

Cambiar dueño y grupo a la vez

sudo chown nuevo_usuario:nuevo_grupo mi_archivo.txt

    - chgrp (Change Group): Cambia solo el grupo. **Puede requerir** sudo.

Bash

sudo chgrp nuevo_grupo mi_archivo.txt

7. Privilegios de Superusuario (Root y sudo**)**

- root**:** Es el superusuario, el administrador total del sistema. Tiene permiso para hacer *cualquier cosa*. Es peligroso trabajar directamente como root porque un error puede dañar gravemente el sistema.
- sudo **(SuperUser DO):** Es la forma recomendada de ejecutar comandos *individuales* con privilegios de root sin necesidad de iniciar sesión como root. Te pedirá *tu* contraseña de usuario (no la de root).

Bash

Intentar instalar software (fallará sin privilegios)

apt install htop # (En Debian/Ubuntu - dará error de permisos)

Intentar lo mismo con sudo (pedirá tu contraseña y funcionará si eres administrador)

sudo apt install htop

Editar un archivo de configuración del sistema (requiere sudo)

sudo nano /etc/hostname

Ver los procesos de todos los usuarios (puede requerir sudo)

sudo ps aux

No todos los usuarios pueden usar sudo. Normalmente, el usuario creado durante la instalación sí puede.

8. Gestión de Paquetes (Instalar Software)

Las distribuciones Linux usan **gestores de paquetes** para instalar, actualizar y eliminar software de forma sencilla y segura desde **repositorios** (servidores que almacenan el software).

- **Debian/Ubuntu (y derivados como Mint):** Usan apt.

Bash

1. Actualizar la lista de paquetes disponibles

sudo apt update

2. Actualizar los paquetes instalados a sus últimas versiones

sudo apt upgrade

3. Buscar un paquete

apt search nombre_paquete

4. Instalar un paquete nuevo

sudo apt install nombre_paquete # ej: sudo apt install gimp

5. Eliminar un paquete

sudo apt remove nombre_paquete

6. Eliminar un paquete y sus archivos de configuración

sudo apt purge nombre_paquete

- **Fedora/CentOS/RHEL (y derivados):** Usan dnf (o yum en versiones más antiguas).

Bash

1. Actualizar paquetes (dnf/yum actualizan lista y paquetes a la vez)

sudo dnf update

o

sudo yum update

2. Buscar un paquete

dnf search nombre_paquete

3. Instalar un paquete nuevo

sudo dnf install nombre_paquete # ej: sudo dnf install gimp

4. Eliminar un paquete

sudo dnf remove nombre_paquete

9. Redirección y Tuberías (Pipes)

Permiten combinar comandos de forma potente.

- **Redirección de Salida (**> **y** >>**):**
    - >: Envía la salida estándar de un comando a un archivo (SOBREESCRIBE el archivo si existe).

Bash

Guarda la lista de archivos en 'lista.txt' (sobrescribe si existe)

ls -l > lista.txt

cat lista.txt

- >>: Envía la salida estándar de un comando a un archivo (AÑADE al final del archivo si existe).

Bash

Añade la fecha actual al final de 'log.txt'

date >> log.txt

cat log.txt

- **Redirección de Entrada (<):** Toma la entrada estándar de un comando desde un archivo (menos común para principiantes).

Bash

Ordenar el contenido de un archivo

sort < lista.txt

- **Tuberías (| - Pipe):** Conecta la salida estándar de un comando con la entrada estándar del siguiente. Permite encadenar comandos.

Bash

Listar todos los archivos, y luego filtrar solo los que contienen '.txt'

ls -a | grep ".txt"

Ver los procesos en ejecución y buscar uno específico (ej: firefox)

ps aux | grep firefox

Contar cuántos archivos hay en el directorio actual

ls | wc -l

10. Búsqueda de Archivos y Texto

- find**:** Busca archivos y directorios según criterios (nombre, tamaño, fecha, tipo, etc.).

Bash

Buscar archivos llamados 'mi_documento.txt' empezando desde el directorio actual (.)

find . -name "mi_documento.txt"

Buscar directorios llamados 'Documentos' empezando desde el home (~)

find ~ -type d -name "Documentos"

Buscar archivos modificados en los últimos 2 días en /etc

find /etc -type f -mtime -2

- grep **(Global Regular Expression Print):** Busca patrones de texto dentro de archivos.

Bash

Buscar la palabra 'error' en un archivo de log

grep "error" /var/log/syslog

Buscar la palabra 'config' (ignorando mayúsculas/minúsculas) en un archivo

grep -i "config" mi_archivo.conf

Buscar líneas que NO contengan 'debug'

grep -v "debug" archivo_log.txt

Buscar recursivamente la palabra 'importante' en todos los archivos dentro de Documentos/

grep -r "importante" Documentos/

11. Conclusión y Próximos Pasos

¡Felicidades! Has cubierto los conceptos básicos más importantes para empezar a usar la terminal de Linux con confianza. Has aprendido a:

- Navegar por el sistema de archivos.
- Crear, copiar, mover y eliminar archivos/directorios.
- Ver el contenido de los archivos.
- Entender y modificar permisos (rwx, chmod).
- Usar privilegios de administrador de forma segura (sudo).

- Instalar y gestionar software (apt/dnf).
- Combinar comandos con redirecciones y tuberías (>, >>, |).
- Buscar archivos y texto (find, grep).

¿Qué sigue?

- **Practica, practica, practica:** La mejor forma de aprender es usando estos comandos.
- **Edición de texto en terminal:** Aprende un editor como nano (simple) o vim/emacs (potentes).
- **Gestión de procesos:** Comandos como ps, top, htop, kill.
- **Redes:** Comandos como ip, ping, ssh, scp.
- **Scripts de Shell:** Automatizar tareas escribiendo tus propios scripts.