

Instructivo Linux

Este documento contiene una guía básica de comandos útiles para trabajar con Linux, organizada por temas y con ejemplos prácticos.

Índice

- [Listar archivos y directorios](#)
 - [Gestión de archivos y directorios](#)
 - [Crear y eliminar directorios](#)
 - [Eliminar archivos](#)
 - [Cambiar de directorio](#)
 - [Permisos de archivos y directorios](#)
 - [Cambiar permisos y propietarios](#)
 - [Ejecutar programas](#)
 - [Redirección de entradas y salidas](#)
 - [Procesos y señales](#)
 - [Buscar texto dentro de archivos](#)
 - [Recortar archivos o líneas](#)
 - [Guardar salida de comandos en variables](#)
 - [Información del sistema](#)
 - [Otros útiles](#)
-

Listar archivos y directorios

`ls`

Parámetros: - `-l`: listado amplio (muestra permisos) - `-a`: muestra todos los archivos (incluyendo ocultos) - `-t`: ordena cronológicamente - `-r`: invierte el orden - `-R`: listado recursivo

Ejemplos:

`ls -ltr`

Alias comunes: - `ls -ll` → `ls -l` - `ls -la` → `ls -a`

Gestión de archivos y directorios

Copiar archivos y carpetas

```
cp archivo1.txt copia.txt
cp -r carpeta1 carpeta2
```

Mover o renombrar archivos

```
mv archivo.txt /otro/directorio/  
mv viejo.txt nuevo.txt
```

Buscar archivos

```
find /ruta -name "*.log"  
find . -type f -size +100M
```

Crear y eliminar directorios

```
mkdir nuevo_directorio  
rmdir directorio_vacío
```

Eliminar archivos

```
rm archivo.txt  
rm -rf carpeta
```

Parámetros: - -f: fuerza la eliminación sin pedir confirmación - -r: recursivo

Cambiar de directorio

```
cd ruta/al/directorio
```

Directorios especiales: - .: actual - ..: superior - ~: home - /: raíz - -: directorio anterior

Permisos de archivos y directorios

Los permisos se representan de dos maneras: texto y números.

Modo texto

```
-rwxr-xr--
```

- 3 posiciones para el usuario, grupo y otros.
- r: lectura, w: escritura, x: ejecución.

Modo numérico

Suma de valores: - 4: lectura - 2: escritura - 1: ejecución

Por ejemplo:

```
chmod 754 archivo
```

Equivale a: - Usuario: 7 (4+2+1) â†’ rwx - Grupo: 5 (4+0+1) â†’ r-x - Otros: 4 (4+0+0) â†’ r--

Cambiar permisos y propietarios

Cambiar permisos

```
chmod 777 archivo  
chmod u+x archivo  
chmod go-wx archivo
```

Cambiar propietario

```
chown usuario archivo  
chgrp grupo archivo
```

Ejecutar programas

- Si estÃ¡ en \$PATH:

```
listar.sh
```

- Si estÃ¡ en el directorio actual:

```
./listar.sh
```

- En background:

```
./listar.sh &  
nohup ./listar.sh &
```

- ComposiciÃ³n de comandos:

```
comando1 | comando2  
comando1 && comando2  
comando1; comando2
```

RedirecciÃ³n de entradas y salidas

```
listar.sh > salida.txt  
listar.sh >> salida.txt  
comando < entrada.txt  
comando < entrada.txt > salida.txt  
borrar.sh 1>log.txt 2>error.txt
```

Procesos y se ales

Ver procesos:

```
ps -f
ps -fu usuario
```

Finalizar procesos:

```
kill -9 PID
kill -SIGSTOP PID
kill -18 PID
```

kill no siempre "mata" un proceso. Env a se ales como:
- SIGKILL (9): mata inmediatamente - SIGTERM (15): solicita cierre amigable - SIGSTOP (19): pausa - SIGCONT (18): contin a

Buscar texto dentro de archivos

```
grep 'texto' archivo.txt
ps -f | grep -i listado
```

Par metros: - -i: ignora may sculas - -v: niega la condici n - -l: muestra nombre de archivos - -n: muestra n mero de l nea - -color: resalta coincidencias

Alternativa avanzada: egrep

Recortar archivos o l neas

```
head -20 archivo.txt
tail -20 archivo.txt
tail -f archivo.log
cut -c1-20 archivo.txt
cut -d';' -f1-3 archivo.txt
```

Guardar salida de comandos en variables

```
VARIABLE=`comando`
```

Informaci n del sistema

```
df -h          # Espacio en disco
du -sh carpeta # Tama o de carpeta
```

```
free -h          # Memoria
uptime          # Tiempo encendido y carga
top             # Procesos activos
```

Otros Ãtiles

Alias

```
alias l='ls -l'
alias actualizar='sudo apt update && sudo apt upgrade'
```

Ayuda

```
man comando
comando --help
```

Shebang en scripts de shell

Al inicio de un script en Linux es comÃn incluir una lÃnea especial llamada **shebang**, que indica quÃ© intÃ©rprete debe usarse para ejecutar el script.

Ejemplo tÃpico para un script de Bash:

```
#!/bin/bash
```

Esto le indica al sistema operativo que debe usar el intÃ©rprete Bash ubicado en /bin/bash para ejecutar el contenido del script.

Otros ejemplos:

```
#!/bin/sh          # Shell POSIX
#!/usr/bin/env python3  # Usa el intÃ©rprete de Python 3 que estÃ© en el e
```

Si un script no tiene shebang, puede ejecutarse directamente con un intÃ©rprete:

```
bash script.sh
```

MÃs informaciÃ³n

- SeÃ±ales: [https://es.wikipedia.org/wiki/SeÃ±al_\(informÃ¡tica\)](https://es.wikipedia.org/wiki/SeÃ±al_(informÃ¡tica))
- Regex: https://es.wikipedia.org/wiki/Expresi%C3%B3n_regular