

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: - `l` → `ls` - `ll` → `ls -l` - `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_vacio
```

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) → `rwx` - Grupo: 5 (4+0+1) → `r-x` -
Otros: 4 (4+0+0) → `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 está en `$PATH`:
`listar.sh`
- Si está en el directorio actual:
`./listar.sh`
- En background:
`./listar.sh &`
`nohup ./listar.sh &`
- Composición de comandos:

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

Redirección 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 señales

Ver procesos:

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

Finalizar procesos:

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

kill no siempre “mata” un proceso. Envía señales como:
- SIGKILL (9): mata inmediatamente - SIGTERM (15): solicita
cierre amigable - SIGSTOP (19): pausa - SIGCONT (18): con-
tinúa

Buscar texto dentro de archivos

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

Parámetros: - -i: ignora mayúsculas - -v: niega la condición - -l:
muestra nombre de archivos - -n: muestra número de línea - --color:
resalta coincidencias

Alternativa avanzada: egrep

Recortar archivos o líneas

```
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`
```

Información del sistema

```
df -h          # Espacio en disco
du -sh carpeta # Tamaño de carpeta
free -h        # Memoria
uptime         # Tiempo encendido y carga
top            # Procesos activos
```

Otros útiles

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 común incluir una línea especial llamada **shebang**, que indica qué intérprete debe usarse para ejecutar el script.

Ejemplo típico para un script de Bash:

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

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

Otros ejemplos:

```
#!/bin/sh           # Shell POSIX
```

```
#!/usr/bin/env python3 # Usa el intérprete de Python 3 que esté en el entorno
```

Si un script no tiene shebang, puede ejecutarse directamente con un intérprete:

```
bash script.sh
```

Más información

- Señales: [https://es.wikipedia.org/wiki/Señal_\(informática\)](https://es.wikipedia.org/wiki/Señal_(informática))
- Regex: https://es.wikipedia.org/wiki/Expresi%C3%B3n_regular