

Directorios Linux

1. / (Raíz)

- **Descripción:** El directorio raíz de todo el sistema de archivos en Linux. Todos los demás directorios y archivos están ubicados bajo este.
- **Contenido:** Subdirectorios que forman la estructura básica del sistema.

2. /bin (Binarios esenciales)

- **Descripción:** Contiene los ejecutables básicos y esenciales del sistema, necesarios para el funcionamiento del mismo, incluso en modo monousuario.
- **Contenido:** Programas como ls, cp, mv, cat, bash, entre otros.

3. /sbin (Binarios de sistema)

- **Descripción:** Similar a /bin, pero contiene programas esenciales que normalmente son utilizados solo por el administrador (root) para gestionar el sistema.
- **Contenido:** Comandos como reboot, shutdown, fdisk, mkfs, etc.

4. /etc (Archivos de configuración)

- **Descripción:** Contiene archivos de configuración global del sistema y los programas instalados. No almacena binarios.
- **Contenido:** Archivos como passwd (para usuarios), fstab (para sistemas de archivos), y otros archivos de configuración de servicios.

5. /home (Directorios de usuarios)

- **Descripción:** Directorios personales de los usuarios del sistema. Cada usuario tiene su propio subdirectorio en /home.
- **Contenido:** Archivos personales, configuraciones, documentos, descargas, etc. Por ejemplo, /home/usuario.

6. /root (Directorio personal de root)

- **Descripción:** Directorio personal del usuario administrador o "root".
- **Contenido:** Similar a /home, pero solo para el usuario root.

7. /lib (Bibliotecas esenciales)

- **Descripción:** Contiene las bibliotecas compartidas necesarias para que los binarios en /bin y /sbin funcionen.
- **Contenido:** Archivos como libc.so.6, que es una biblioteca esencial en muchos sistemas.

8. /usr (Archivos del usuario)

- **Descripción:** Contiene programas y archivos de uso general por los usuarios y administradores, pero que no son esenciales para el arranque del sistema.
- **Subdirectorios importantes:**
 - **/usr/bin:** Ejecutables de usuario (no esenciales).
 - **/usr/sbin:** Ejecutables de sistema no esenciales.
 - **/usr/lib:** Bibliotecas compartidas para /usr/bin y /usr/sbin.
 - **/usr/share:** Archivos compartidos como documentación y datos.
 - **/usr/local:** Para software instalado manualmente por el administrador, fuera de los paquetes del sistema.

9. /var (Archivos variables)

- **Descripción:** Contiene archivos y datos que cambian con el tiempo, como logs, cachés, y archivos temporales del sistema.
- **Contenido:**
 - **/var/log:** Archivos de registro del sistema, como syslog, auth.log.
 - **/var/cache:** Datos almacenados en caché por aplicaciones.
 - **/var/spool:** Archivos en cola para procesar, como trabajos de impresión o correo.

10. /tmp (Archivos temporales)

- **Descripción:** Almacena archivos temporales creados por aplicaciones y el sistema.
- **Contenido:** Archivos que pueden ser eliminados automáticamente después de reiniciar el sistema o después de un tiempo.

11. /dev (Dispositivos)

- **Descripción:** Contiene archivos especiales que representan dispositivos de hardware del sistema, como discos duros, particiones, y periféricos.
- **Contenido:** Archivos de dispositivos como /dev/sda (disco duro), /dev/null (dispositivo nulo), /dev/tty (terminales).

12. */mnt y /media (Puntos de montaje)

- **Descripción:** Lugares donde se montan sistemas de archivos temporales, como dispositivos USB o particiones adicionales.
- **Contenido:** Usados para montar discos duros externos, CD-ROMs, o sistemas de archivos de red.

13. /proc (Información del sistema y procesos)

- **Descripción:** Sistema de archivos virtual que proporciona información sobre los procesos y el estado del kernel.

- **Contenido:** Archivos como `/proc/cpuinfo`, `/proc/meminfo`, que muestran detalles sobre el procesador y la memoria.

14. `/sys` (Información del sistema)

- **Descripción:** Sistema de archivos virtual similar a `/proc`, que proporciona información y acceso a dispositivos y buses del sistema.
- **Contenido:** Información sobre dispositivos y controladores de hardware.

15. `/boot` (Archivos de arranque)

- **Descripción:** Contiene los archivos necesarios para iniciar el sistema, incluido el núcleo (kernel) y el cargador de arranque.
- **Contenido:** Archivos como `vmlinuz` (el kernel de Linux) y `grub` (archivos del gestor de arranque).

16. `/opt` (Software opcional)

- **Descripción:** Directorio donde se instalan aplicaciones adicionales o de terceros que no forman parte del sistema base.
- **Contenido:** Paquetes de software que no se instalan mediante el gestor de paquetes del sistema.

17. `/srv` (Datos de servicios)

- **Descripción:** Contiene datos para servicios ofrecidos por el sistema, como servidores web o FTP.
- **Contenido:** Archivos para servidores como páginas web (`/srv/www`), datos FTP, etc.

Resumen:

- **`/bin`, `/sbin`, `/usr/bin`, `/usr/sbin`:** Ejecutables del sistema.
- **`/etc`:** Archivos de configuración.
- **`/home`, `/root`:** Directorios personales de usuarios.
- **`/lib`, `/usr/lib`:** Bibliotecas compartidas.
- **`/var`:** Datos variables, como logs y cachés.
- **`/tmp`:** Archivos temporales.
- **`/dev`, `/proc`, `/sys`:** Información y control de dispositivos y procesos.
- **`/boot`:** Archivos de arranque del sistema.
- **`/opt`:** Software opcional de terceros.

Cada uno de estos directorios cumple una función específica para la estructura organizada y eficiente del sistema operativo Linux.

Operadores Linux

1. Operadores Aritméticos

Estos operadores permiten realizar operaciones matemáticas con enteros.

- **+**: Suma ($((a + b))$)
- **-**: Resta ($((a - b))$)
- *****: Multiplicación ($((a \ * \ b))$)
- **/**: División ($((a / b))$)
- **%**: Módulo (resto de la división) ($((a \% b))$)
- ******: Potencia (no soportado directamente en $(())$, pero se puede hacer con otras herramientas como bc)

2. Operadores de Comparación Numérica

Se utilizan para comparar valores numéricos.

- **-eq**: Igual a ($[a -eq b]$)
- **-ne**: Distinto de ($[a -ne b]$)
- **-lt**: Menor que ($[a -lt b]$)
- **-le**: Menor o igual a ($[a -le b]$)
- **-gt**: Mayor que ($[a -gt b]$)
- **-ge**: Mayor o igual a ($[a -ge b]$)

3. Operadores de Comparación de Cadenas

Comparan cadenas de texto.

- **=**: Igual a ($["$cadena1" = "$cadena2"]$)
- **!=**: Distinto de ($["$cadena1" != "$cadena2"]$)
- **-z**: Longitud de la cadena es 0 ($[-z "$cadena"]$)
- **-n**: Longitud de la cadena no es 0 ($[-n "$cadena"]$)

4. Operadores Lógicos

Se usan para combinar condiciones.

- **&&**: AND lógico (evalúa si ambas expresiones son verdaderas) ($[a -gt 0] \&\& [b -gt 0]$)
- **||**: OR lógico (evalúa si al menos una de las expresiones es verdadera) ($[a -gt 0] || [b -gt 0]$)

- **!:** NOT lógico (invierte el resultado de la expresión) ([! -f archivo])

5. Operadores de Archivos

Se utilizan para verificar diferentes propiedades de archivos.

- **-e archivo:** Verifica si el archivo existe.
- **-f archivo:** Verifica si es un archivo regular.
- **-d archivo:** Verifica si es un directorio.
- **-r archivo:** Verifica si el archivo tiene permisos de lectura.
- **-w archivo:** Verifica si el archivo tiene permisos de escritura.
- **-x archivo:** Verifica si el archivo tiene permisos de ejecución.

6. Operadores de Asignación

Se usan para asignar valores a variables.

- **=:** Asignación de valor (var=valor)
- **+=:** Suma y asigna (a+=1)
- **-=:** Resta y asigna (a-=1)

7. Operadores de Concatenación de Cadenas

Concatenan cadenas de texto.

- **cadena1="\$cadena1\$cadena2":** Concatenación (cadena1="Hola", cadena2="Mundo", resultado: cadena1="Hola Mundo").

8. Operadores de Redirección

Se utilizan para redirigir la entrada y salida.

- **>:** Redirige la salida a un archivo (sobrescribe).
- **>>:** Redirige la salida a un archivo (agrega al final).
- **<:** Redirige la entrada de un archivo.
- **2>:** Redirige la salida de error a un archivo.

find, tr, grep

find

El comando find se utiliza para buscar archivos y directorios en una estructura de directorios. Es poderoso y permite búsquedas basadas en múltiples criterios.

- **Sintaxis básica:**

`find [ruta] [opciones]`

- **Opciones comunes:**

- `-name <nombre>`: Busca archivos o directorios por nombre.
 - Ejemplo: `find /home -name "*.txt"` busca todos los archivos .txt en /home.
- `-type <tipo>`: Busca un tipo específico (f para archivo, d para directorio).
 - Ejemplo: `find /home -type d` busca solo directorios.
- `-size <n>`: Busca archivos según su tamaño.
 - Ejemplo: `find /var -size +10M` busca archivos de más de 10 MB.
- `-user <usuario>`: Busca archivos que pertenezcan a un usuario específico.
 - Ejemplo: `find /home -user nombreusuario`.
- `-mtime <n>`: Busca archivos modificados hace n días.
 - Ejemplo: `find /home -mtime -7` busca archivos modificados en la última semana.
- `-exec <comando>`: Ejecuta un comando para cada archivo encontrado.
 - Ejemplo: `find /home -name "*.log" -exec rm {} \;` elimina todos los archivos .log.

2. tr

El comando tr (translate) se usa para transformar o eliminar caracteres en un flujo de texto.

- **Sintaxis básica:**

`tr [opciones] set1 [set2]`

- **Opciones comunes:**

- `tr 'a-z' 'A-Z'`: Convierte letras minúsculas a mayúsculas.
- `tr -d 'a'`: Elimina todas las apariciones del carácter a.

- **tr ' ' '\n':** Convierte espacios a saltos de línea, separando cada palabra en una línea diferente.
- **tr -s ' ':** Reemplaza múltiples espacios por un único espacio (compacta espacios).
- **Ejemplo práctico:**

```
echo "Hola Mundo" | tr 'a-z' 'A-Z'
```

Salida: HOLA MUNDO (convierte las letras minúsculas a mayúsculas).

3. grep

El comando grep se utiliza para buscar patrones en archivos o en la entrada estándar. grep es ideal para filtrar líneas que coinciden con una expresión regular.

- **Sintaxis básica:**

```
grep [opciones] 'patrón' [archivo]
```

- **Opciones comunes:**
 - **-i:** Ignora mayúsculas y minúsculas en la búsqueda.
 - Ejemplo: `grep -i 'error' archivo.log` busca 'error', 'Error', etc.
 - **-v:** Muestra las líneas que **no** coinciden con el patrón.
 - Ejemplo: `grep -v 'ERROR' archivo.log` muestra las líneas que no contienen 'ERROR'.
 - **-r:** Busca recursivamente en directorios.
 - Ejemplo: `grep -r 'TODO' /home/proyecto` busca 'TODO' en todos los archivos del directorio /home/proyecto.
 - **-n:** Muestra los números de línea donde se encuentran las coincidencias.
 - Ejemplo: `grep -n 'ERROR' archivo.log`.
 - **-w:** Coincidencia exacta de palabras.
 - Ejemplo: `grep -w 'cat' archivo.txt` encuentra 'cat' pero no 'catch'.

4. ls

El comando ls se utiliza para listar archivos y directorios en la terminal. Es uno de los comandos más básicos y útiles para ver el contenido de un directorio.

- **Sintaxis básica:**

```
ls [opciones] [ruta]
```

- **Opciones comunes:**
 - **-l:** Muestra detalles en formato largo (permisos, propietario, tamaño, fecha, etc.).
 - Ejemplo: `ls -l /home`.

- -a: Muestra todos los archivos, incluidos los archivos ocultos (que comienzan con .).
 - Ejemplo: `ls -a`.
- -la: Combina las opciones -l y -a, mostrando detalles de todos los archivos, incluidos los ocultos.
- -h: Formato legible para humanos (muestra tamaños como 1K, 1M, etc.).
 - Ejemplo: `ls -lh`.
- -R: Lista el contenido de los subdirectorios de manera recursiva.
 - Ejemplo: `ls -R /home`.
- -t: Ordena los archivos por la fecha de modificación.
 - Ejemplo: `ls -lt`.

Tar, gzip, ps

Comando tar

tar (Tape Archive) se utiliza para crear, extraer y manipular archivos comprimidos (archivos con extensión .tar). Es útil para agrupar varios archivos o directorios en uno solo, sin comprimirlos. Puedes combinar tar con gzip para crear archivos .tar.gz, que son comprimidos.

Sintaxis básica:

```
tar [opciones] [archivo.tar] [archivos o directorios]
```

Opciones comunes:

1. **-c:** Crear un archivo.
2. **-x:** Extraer un archivo.
3. **-v:** Modo detallado (muestra archivos mientras se procesan).
4. **-f:** Especifica el nombre del archivo .tar.
5. **-z:** Comprime o descomprime con gzip (crea archivos .tar.gz).

Ejemplos:

1. **Crear un archivo .tar** (sin compresión):

```
tar -cvf archivo.tar directorio/
```

Esto agrupa el contenido del directorio en un archivo llamado archivo.tar.

2. **Crear un archivo .tar.gz** (con compresión gzip):

```
tar -czvf archivo.tar.gz directorio/
```

Esto agrupa y comprime el directorio en un archivo archivo.tar.gz.

3. **Extraer un archivo .tar:**

```
tar -xvf archivo.tar
```

Extrae el contenido de archivo.tar en el directorio actual.

4. Extraer un archivo .tar.gz:

```
tar -xvzf archivo.tar.gz
```

Descomprime y extrae archivo.tar.gz en el directorio actual.

Comando gzip

gzip es una utilidad de compresión de archivos que reduce el tamaño de un archivo. Crea archivos con extensión .gz. No es un archivo de archivado como tar, ya que comprime archivos individuales, no múltiples.

Sintaxis básica:

```
gzip [opciones] [archivo]
```

Opciones comunes:

1. **-d:** Descomprime un archivo .gz.
2. **-k:** Mantiene el archivo original después de la compresión.
3. **-r:** Comprime archivos recursivamente en un directorio.

Ejemplos:

1. Comprimir un archivo:

```
gzip archivo.txt
```

Esto comprime archivo.txt a archivo.txt.gz.

2. Descomprimir un archivo:

```
gzip -d archivo.txt.gz
```

Esto descomprime archivo.txt.gz a archivo.txt.

3. Comprimir todos los archivos en un directorio:

```
gzip -r directorio/
```

Comprime todos los archivos en el directorio (reemplaza cada archivo con su versión .gz).

Comando ps

ps (Process Status) muestra información sobre los procesos que están ejecutándose en el sistema.

Sintaxis básica:

`ps [opciones]`

Opciones comunes:

1. **-e o -A:** Muestra todos los procesos del sistema.
2. **-f:** Muestra información detallada.
3. **-u [usuario]:** Muestra los procesos de un usuario específico.
4. **aux:** Es una combinación común que muestra todos los procesos en detalle.

Ejemplos:

1. **Listar los procesos del usuario actual:**

`ps`

Muestra solo los procesos del usuario que ejecuta el comando.

2. **Listar todos los procesos del sistema:**

`ps -e`

Muestra una lista de todos los procesos en ejecución.

3. **Listar todos los procesos en formato detallado:**

`ps -ef`

Muestra todos los procesos con información detallada (PID, usuario, CPU, memoria, etc.).

4. **Buscar un proceso específico usando grep:**

`ps -ef | grep nombre_del_proceso`

Filtra la salida para encontrar procesos específicos.