

Usando el comando Find en Linux

El comando **find** es el comando que más se utiliza para encontrar y filtrar archivos en Linux. El diseño básico de este comando es el siguiente:

```
find <startingdirectory> <options> <search term>
```

Comienza con la palabra clave **find**, que alerta a Linux de que lo que sigue se refiere a la búsqueda de un archivo. El argumento **<startingdirectory>** es el punto de origen de donde deseas iniciar la búsqueda. Puede ser reemplazado con varios argumento, incluyendo:

- **/ (slash)** – busca en todo el sistema.
- **. (punto)** – busca en la carpeta en la que estás trabajando actualmente (directorio actual).
- **~ (tilde)** – para buscar desde tu directorio home.

Para averiguar tu directorio actual, usa el comando **pwd**.

El segundo argumento **<options>** se usa para tu archivo. Este podría ser el nombre, tipo, fecha de creación del archivo, etc. El tercer argumento **<searchterm>** es donde se especificará el término de búsqueda relevante.

Búsqueda por nombre

Por supuesto, el método más común y obvio para buscar un archivo es usar su nombre. Para ejecutar una consulta de búsqueda simple usando el nombre del archivo, usa el comando **find** de la siguiente manera:

```
find . -name my-file
```

Usamos la opción **-name** y buscamos un archivo llamado **my-file**. Ten presente que comenzamos la búsqueda en nuestro directorio actual usando el argumento **.** (**punto**).

Recuerda que el argumento **-name** busca términos distinguiendo entre mayúsculas y minúsculas en Linux. Si conoces el nombre del archivo, pero no estás seguro de su las mayúsculas y minúsculas, usa el comando **find** de esta manera:

```
find . -iname my-file
```

También puedes buscar todos los archivos sin una palabra clave en el nombre. Puedes hacer esto de dos maneras. El primer método implica el uso de la palabra clave **-not** de la siguiente manera:

```
find . -not -name my-file
```

En el segundo, podemos usar el signo de exclamación (!), aunque debe estar precedido por el identificador de escape (\) para que Linux sepa que este es parte del comando **find**.

```
find . \! -name my-file
```

También puedes buscar varios archivos con un formato común como **.txt**:

```
find . -name "*.txt"
```

Esto listará todos los archivos de texto comenzando con la carpeta actual.

Finalmente, si deseas buscar un determinado archivo por nombre y eliminarlo, usa el argumento **-delete** después del nombre del archivo:

```
find . -name my-file -delete
```

Búsqueda por tipo

Linux permite a los usuarios listar toda la información basada en sus tipos. Hay varios filtros que puedes usar:

- **d** – directorio o carpeta
- **f** – archivo normal
- **l** – enlace simbólico
- **c** – dispositivos de caracteres
- **b** – dispositivos de bloque

Un ejemplo simple del uso del tipo de archivo para la búsqueda se puede ver a continuación:

```
find / -type d
```

Esto mostrará una lista de todos los directorios presentes en tu sistema de archivos, al haber comenzado la búsqueda desde nuestro directorio raíz con el símbolo de barra inclinada `/`.

También puedes concatenar las opciones **-type** y **-name** para hacer tus búsquedas aún más específicas:

```
find / -type f -name my-file
```

Esto buscará archivos llamados **my-file**, excluyendo directorios o enlaces.

Búsqueda por fecha

Si quieres buscar archivos en función de su fecha de acceso y los registros de fecha de modificación, Linux te ofrece las herramientas para hacerlo. Hay 3 registros de tiempo de los cuales Linux realiza seguimiento en los archivos:

- **Tiempo de acceso (-atime)** – Fecha más reciente en que el archivo fue leído o escrito.
- **Tiempo de modificación (-mtime)** – Fecha más reciente en que se modificó el archivo.
- **Hora de cambio (-ctime)** – Fecha más reciente en que se actualizaron los metadatos del archivo.

Esta opción debe usarse con un número que especifica cuántos días pasaron desde que se accedió, modificó o cambió el archivo:

```
find / -atime 1
```

Este comando mostrara todos los archivos a los que se accedió hace un día desde el momento actual.

Podemos hacer que nuestras consultas sean más precisas agregando los signos **más (+)** y **menos (-)** precediendo al número de días. Por ejemplo:

```
find / -mtime +2
```

Esto listará todos los archivos que tienen un tiempo de modificación de más de dos días.

Para buscar todos los archivos cuyos metadatos se actualizaron hace menos de un día, ejecuta lo siguiente:

```
find / -ctime -1
```

Aunque no se usan seguido, hay algunos argumentos adicionales que también están relacionados con las búsquedas por fecha. El argumento **-mmin** busca archivos modificados en base a minutos. Se puede usar así:

```
find / -mmin -1
```

Además, el argumento **-newer** se puede usar para comparar la antigüedad de dos archivos y encontrar el más reciente.

```
find / -newer my-file
```

Obtendrás todos los archivos que han sido modificados hace menos tiempo que tu archivo.

Búsqueda por tamaño

Linux te brinda la opción de buscar archivos según sus tamaños. La sintaxis básica para buscar archivos por tamaño es:

```
find <startingdirectory> -size <size-magnitude> <size-unit>
```

Puedes especificar las siguientes unidades de tamaño:

- **c** – bytes
- **k** – kilobytes
- **M** – megabytes
- **G** – gigabytes
- **b** – trozos de 512 bytes

Un ejemplo simple de cómo usar el comando **find** de Linux para los tamaños de archivo es el siguiente:

```
find / -size 10M
```

Esto buscará en tu sistema archivos que tengan exactamente 10 megabytes de tamaño. Al igual que cuando buscaste en función del tiempo, puedes filtrar aún más tus búsquedas con los signos más y menos:

```
find / -size +5G
```

El comando anterior listará todos los archivos de tu disco que tengan más de 5 Gigabytes de tamaño.

Búsqueda por propiedad

Linux te da la capacidad de especificar tus búsquedas según la propiedad del archivo. Para buscar archivos de un determinado propietario, se debe ejecutar el siguiente comando:

```
find / -user john
```

Esto devolverá una lista de todos los archivos que posee el usuario llamado **john**. Similar a los nombres de usuario, también podemos buscar archivos a través de nombres de grupo:

```
find / -group classroom
```

Búsqueda por permisos

Los usuarios pueden buscar archivos basados en los permisos de los archivos con la opción **-perm**. Por ejemplo:

```
find / -perm 644
```

En Linux, **644** corresponde a permisos de lectura y escritura. Lo que significa que este comando buscará todos los archivos que solo tienen permisos de lectura y escritura. Puedes jugar con esta opción un poco más, así:

```
find / -perm -644
```

Al agregar un guión, se mostrarán todos los archivos que tengan al menos el permiso 644.

Otras opciones útiles

Además de todos estos métodos de búsqueda de archivos, hay otras opciones útiles que deberías recordar.

Por ejemplo, para buscar archivos y carpetas vacíos en tu sistema, usa lo siguiente:

```
find / -empty
```

Del mismo modo, para buscar todos los ejecutables guardados en tu disco, utiliza la opción **-exec**:

```
find / -exec
```

Para buscar archivos legibles, puedes ejecutar el siguiente comando:

```
find / -read
```