

# Herramientas para Localizar Ficheros

Lo más habitual para localizar ficheros es hacerlo por su nombre, pero a veces se pueden localizar por otros criterios, como por ejemplo la fecha de modificación. Los comandos de localización de ficheros de Linux pueden buscar en un árbol de directorios un fichero que coincida con el criterio especificado e incluso podrían rastrear el sistema completo.

## El Comando find

El comando **find** implementa un sistema de fuerza bruta para localizar ficheros ya que realiza una búsqueda por todo el árbol de directorios especificado para localizar los que satisfagan el criterio especificado. **find** tiende a ser lento pero es flexible y tiene una mayor probabilidad de éxito.

```
find [ruta...] [expresión...]
```

Se le pueden especificar una o más rutas para restringir sus operaciones. La expresión es un modo de especificar lo que queremos buscar; las expresiones más comunes permiten buscar por varios criterios:

- Buscar por nombre de fichero: Utilizando la expresión **-nombre patrón** se buscarán los ficheros que coincidan con el patrón especificado. Se pueden emplear comodines delimitando el patrón entre comillas.
- Buscar por el modo de permiso: La expresión **-perm modo** permite localizar ficheros que tengan ciertos permisos. El modo se puede expresar simbólicamente o en modo octal si va precedido de un mas (+). El comando **find** localizará los ficheros en los que estén definidos todos los bits de permisos especificados.
- Buscar por el tamaño del fichero: Se puede realizar una búsqueda de un fichero de un tamaño determinado con la expresión **-size n**, donde "n" especifica en bloques de 512 bytes (se puede modificar con una letra que sirva de código para el valor, por ejemplo, k para kilobytes).
- Buscar por grupo: La expresión **-gid GID**, busca los ficheros cuya ID de grupo es GID. La opción **-group nombre** localiza ficheros cuyo grupo tiene el nombre indicado. Esta última opción suele ser más fácil de utilizar pero la primera es útil en caso de que la GID se haya quedado huérfana y no tenga nombre.
- Buscar por ID de usuario: **-uid UID** busca los ficheros cuyo propietario tiene UID como ID de usuario. **-user nombre** busca los ficheros cuyo propietario es nombre. Esta última opción suele ser más fácil de utilizar pero la primera es útil en caso de que la UID se haya quedado huérfana y no tenga nombre.
- Restringir la profundidad de la búsqueda: Para limitar el número de subdirectorios en los que buscar se emplea la expresión **-maxdepth niveles**.

**find** tiene muchas variantes y opciones adicionales, pero por ejemplo si se desean localizar todos los ficheros fuente en código #C con nombres que normalmente acaban en .c y se encuentran el directorio *home* , se ejecutará:

```
find /home -name '*.c'
```

Esta expresión devolverá todos los ficheros que coincidan con el criterio de búsqueda.

Los usuarios normales pueden utilizar **find**, pero si carecen de permisos para listar el contenido de un directorio, **find** devolverá un error junto con el nombre del directorio.

## El Comando locate

Esta utilidad funciona de forma similar a **find** pero difiere en dos aspectos importantes:

- Es mucho menos sofisticado en sus opciones de búsqueda ya que se utiliza para buscar por el nombre de fichero con lo que el programa devuelve todos los ficheros que contienen la cadena especificada.
- Trabaja con una base de datos que se encarga de mantener. Normalmente, se incluye una tarea cron que llama a **locate** para que actualice su base de datos periódicamente. El comando **updatedb** sirve para actualizar la base de datos manualmente. Es importante que se mantenga actualizada, ya que en caso contrario, al realizar las búsquedas puede que devuelva nombres de ficheros que ya no existan o que no localice ficheros recientes.

Al trabajar con una base de datos es más rápido que **find** , sobre todo en las búsquedas por todo el sistema. También es probable que devuelva muchas falsas alarmas, especialmente si se quiere localizar un fichero con un nombre corto. Algunas distribuciones utilizan **slocate** en vez de **locate**, que incluye características de seguridad que impiden que los usuarios vean los nombres de los ficheros o de los directorios a los que no tienen acceso. En la mayoría de los sistemas con **slocate**, **locate** es un enlace a **slocate**.