

Jun 7, 2013

Comando find

By mar10

- [Inicie sesión](#) o [regístrese](#) para comentar
- 2541 lecturas

Es el caso del comando find,

Uno de los legendarios de los sistemas Unix y por supuesto de las distribuciones Linux, y que permite realizar todo tipo de búsquedas de ficheros tanto por los nombre como por los distintos atributos que tienen esos mismos ficheros. En Linux.com han realizado un práctico tutorial con 10 consejos para usar find, y lo cierto es que este recorrido nos permitirá a buen seguro expresar el funcionamiento de este comando. 1. Uso básico Lo normal es que utilicemos find para encontrar el nombre de un fichero en el directorio, no solo donde lo dejamos, sino cómo se llama ese fichero exactamente. Para ello basta con utilizar el formato

```
find -name nombrefichero
```

2. Búsqueda por tamaño A veces es importante encontrar ficheros por su tamaño, y para ello basta aplicar un parámetro:

```
find -size 100k
```

lo que permitiría buscar ficheros con un tamaño exacto de 100 kilobytes. Es probable que lo que queramos es encontrar ficheros con un tamaño superior o inferior a esa cifra en cuyo caso usaríamos

```
find size +100k
```

para los ficheros con mayor tamaño de 100k, y

```
find size -100
```

para ficheros de menor de 100k. Por cierto, si necesitamos buscar en gigabytes, basta con poner 100G, por ejemplo. 3. Búsqueda por usuario También podemos buscar ficheros que pertenecen a cierto usuario... o que no pertenecen a nadie y han quedado "huérfanos", con la orden:

```
find -nouser
```

Si queremos buscar ficheros de alguien concreto usaremos las opciones -user o -uid. En este último caso deberemos saber el identificador de un usuario, algo que no es lo normal que sepamos, pero siempre viene bien tenerlo en cuenta. Además podemos buscar ficheros que pertenecen a un usuario A o a un usuario B con el parámetro -o, con lo que escribiríamos:

```
find -user root -o -user www-data
```

Y como era de esperar, también podemos buscar los ficheros que NO pertenecen a un usuario concreto:

```
find -not -user www-data
```

4. Búsqueda por grupo Lo mismo se podría aplicar a la búsqueda por grupos que puede afectar al nombre de grupo o al GID de ese grupo (con el mismo flag, cuidado), y haríamos un simple

```
find -group nombredelgrupo
```

(o gid del grupo) Obviamente, podemos combinar esta opción con el resto de opciones, para por ejemplo encontrar ficheros que pertenecen a cierto grupo y que además pertenecen a cierto usuario dentro del grupo.

5. Búsqueda por permisos Uno de los usos más importantes de find -y probablemente de los más desconocidos- es su capacidad de encontrar ficheros con ciertos permisos, algo que nos permite detectar archivos que no tienen los permisos que deberían tener y que pueden suponer una amenaza para la seguridad del sistema. Las opciones más sencillas de usar son -readable, -writable y -executable, aunque también podemos usar el parámetro -perm a partir del cual podemos establecer esos tres permisos a la vez con la típica combinación de tres números. Por ejemplo:

```
find -type f -perm -110
```

Buscaría ficheros con el bit de ejecutable activado para el usuario y el grupo, y el "-" le indica que ignore otros bits. O sea, que no importa si el fichero es de lectura o de escritura, porque lo que queremos es que aparezcan todos los que tienen ese bit de ejecución. Si queremos que específicamente se busquen ficheros que tengan el bit ejecutable para usuario y grupo usaríamos "/", y no "-":

```
find -type f -perm /110
```

Lo que buscaría solo aquellos ficheros con esos permisos, y no otros. 6. Usando expresiones regulares La potencia del comando find crece al usar las expresiones regulares (con la sintaxis tradicional regex) que permite aplicar estas potentes (pero complejas) expresiones para realizar búsquedas mucho más avanzadas. Podemos establecer este tipo de búsqueda con el parámetro -regextype, y luego alguno de los tipos de expresiones soportados con los parámetros posix-awk, posix-egrep, etc. Consultad la página del manual para ampliar información. Por ejemplo si quisiéramos encontrar ficheros con extensión .php o .js haríamos lo siguiente:

```
find -regextype posix-egrep -regex '.*(php|js)$'
```

Al principio puede parece un poco complicado, pero si analizáis la expresión regular todo cobra sentido. La potencia que ofrecen las expresiones regulares es impresionante, pero ya solo este tema bastaría para un post dedicado a ellas. 7. Búsquedas por fechas Si quisiéramos encontrar ficheros por su edad, podríamos hacerlo también con tres parámetros específicos para estas tareas. Por tiempo de acceso (-atime), por última modificación (mtime) o por el último cambio de fichero (-ctime). Todas las cifras, expresadas en días, cuidado. Por ejemplo, busquemos ficheros que no hayan sido modificados en los dos últimos días

```
find -mtime +2
```

Podemos combinar las opciones y buscar ficheros en un rango, por ejemplo, ficheros que tengan entre 2 y 5 días de edad:

```
find -mtime +2 -mtime -5
```

8. Búsquedas por minutos Una especie de rama de lo anterior, pero que es importante porque trabaja con sus propios parámetros que son -amin, -cmin y -mmin, análogas a las que habíamos visto antes. Simplemente hay que tener en cuenta que en este caso la búsqueda es de cifras en minutos, no de días, lo que permite buscar con mucha mayor precisión en el tiempo. 9. Limitar las búsquedas A veces el comando find proporciona más resultados de los que necesitamos o queremos, y para limitar esos resultados podemos utilizar la opción -maxdepth. Por ejemplo, si queremos encontrar un archivo JavaScript en nuestro directorio wordpress, haríamos lo siguiente:

```
find wordpress -name "*.js"
```

pero si solo quisiéramos buscar en el directorio de primer nivel haríamos

```
find wordpress -maxdepth 1 -name "*.js"
```

Si quisiéramos buscar en el directorio y en la primera rama de subdirectorios habría que poner "-maxdepth 2", y así sucesivamente según la profundidad de directorios en la que queremos buscar ficheros. 10. Combinando find con otros comandos Como era de esperar, es posible combinar find con otros comandos para construir potentes sentencias, por ejemplo en conjunción con xargs o con la opción -exec para manipular los ficheros

tras haberlos encontrado. Por ejemplo, si quisiéramos cambiar los permisos de un puñado de ficheros de root al usuario www-data haríamos:

```
find -user root -exec chown www-data {} \;
```

Este comando hace que todos los ficheros que se encuentren y sean propiedad de root, sean utilizados por el comando chown para cambiarle el propietario. Como suele decirse, las posibilidades parecen ilimitadas. Pero como veis, find es un comando espectacularmente útil si se sabe usar con sabiduría.

[FUENTE ORIGINAL](#)

Fruteroloco by Linux