

Buscar archivos

1. Consideraciones generales

El comando **find** permite buscar archivos dentro de la estructura del sistema de archivos con la ayuda de criterios, y da la posibilidad de actuar sobre los resultados devueltos.

```
find ruta criterios opciones
```

Al ser recursivo el comando **find**, basta con indicar un directorio básico para que se desarrolle toda la estructura desde este directorio. La opción básica es `-print` (a menudo implícita en la mayoría de los Unix), que permite visualizar en pantalla los resultados.

```
[jolivares@client slyunix]$ find
.
./logos-cuadrado.tif
./logos-cuadrado.eps
./Pagina 5.pdf
./logos-cuadrado-grande.jpg
./LOGOS
./sitio_2.jpg
./pub_planeta.pdf
./index_logon_inc.php
./logo-iceberg.eps
./flyer
...
```

La visualización es relativa, ya que la ruta indicada es relativa. Si se hubiese especificado la ruta absoluta, la visualización habría sido absoluta.

2. Criterios de búsqueda

Los parámetros permiten definir los criterios de búsqueda. Si hay varios, puede combinarlos usando una Y lógica (criterio1 Y criterio2).

a. -name

`-name` permite una selección por nombres de archivos. Es posible utilizar los comodines ya vistos. Se coloca el criterio entre comillas. Aquí se visualiza la lista de todos los archivos desde la ubicación actual y que empiezan por "fic".

```
$ find . -name "fic*" -print
./fic1
./fic2
./fic3
./fic4
```

b. -type

`-type` permite una selección por tipo de archivos. Ya sabe que, además de los vínculos, los directorios y los archivos simples, están presentes otros tipos de archivos.

Código	Tipo de archivo
b	Archivo especial en modo bloque
c	Archivo especial en modo carácter
d	Directorio (directory)
f	Archivo ordinario
l	Vínculo simbólico
p	Tubería con nombre (pipe)
s	Socket (Conexión de red)

Se visualizan todos los directorios cuyo nombre empieza por "re".

```
$ find . -name "re*" -type d -print
./dir1
./dir2
```

c. -user y -group

`-user` y `-group` permiten una búsqueda sobre el propietario y el grupo de pertenencia de los archivos. Es posible precisar el nombre (usuario, grupo) o la ID (UID, GID). El ejemplo siguiente busca todos los archivos ordinarios que pertenecen a `seb` y al grupo `users`.

```
$ find . -type f -user jolivares -group users -print
./fic1
./fic3
```

d. -size

`-size` permite precisar el tamaño de los archivos buscados. Su sintaxis es particular, ya que trabaja por defecto en bloques si no se especifica nada. Puede resultar sorprendente a veces; más aún cuando el bloque, que aquí tiene un tamaño de 512 bytes, es una unidad un poco virtual (con ciertos comandos, un bloque puede ser de 1 KB o más).

Los caracteres b, c, w o k pueden aparecer tras el valor colocado después del criterio.

Carácter	Significado
b	Por defecto, si no se especifica, es un bloque de 512 bytes.
c	Es un carácter en el sentido ASCII; por lo tanto, 1 byte.
w	Es una palabra (en el sentido antiguo) de 2 bytes.
k	1 KB (1024 bytes).
M	1 MB (1024 KB)
G	1 GB (1024 MB)

El valor puede ir precedido de un + o un - que significan «más de» o «menos de». Sin esta indicación, el tamaño buscado debe corresponderse EXACTAMENTE.

~ -size 5: busca los archivos de un tamaño de 5 bloques (512 bytes por bloque, o

sea, 2560 bytes).

- ✓ -size 152c: busca los archivos de un tamaño de 152 caracteres (bytes).
- ✓ -size 10k: busca los archivos de un tamaño de 10 KB (10*1024 bytes = 10240 bytes).
- ✓ -size +5000k: los archivos de más de 5000 KB.
- ✓ -size -100k: los archivos de menos de 100 KB.

```
seb@slyserver:/var/log> find -size +100k
./zypper.log-20080227.bz2
./lastlog
./zypper.log-20080302.bz2
./wtmpt
./zypper.log-20080226.bz2
./zypper.log
./messages
```



Se puede utilizar el criterio de búsqueda **-empty** en sustitución de -size 0.

e. -atime, -mtime y -ctime

- ✓ **-atime**: busca en la fecha del último acceso (access time). Un acceso puede ser la lectura del archivo, pero también el simple hecho de listarlo de manera específica.
- ✓ **-mtime**: busca en la fecha de la última modificación (modification time). Se trata de la modificación del contenido.
- ✓ **-ctime**: busca en la fecha de modificación (change time, en realidad la fecha de última modificación del número de inodo).



La fecha de modificación del archivo corresponde a la fecha en la cual se modificó por última vez la información relacionada con el inodo (ver capítulo Los discos y el sistema de archivos): modificación del nombre, desplazamiento, cambio de derechos, del tamaño, etc. Observe que POSIX no dispone de información acerca de la fecha de creación de un archivo.

Estos tres criterios sólo trabajan con días (periodos de 24 horas). 0 es el mismo día; 1, ayer; 2, antes de ayer, etc. El valor n colocado después del criterio corresponde, por lo tanto, a $n \times 24$ horas. Este rango no es fijo, ya que «ayer» significa entre 24 y 48 horas...

Los signos + o - permiten precisar los términos "de más" y "de menos":

- ~ -mtime 1: archivos modificados ayer (entre 24 y 48 horas).
- ~ -mtime -3: archivos modificados hace menos de tres días (72 horas).
- ~ -atime +4: archivos a los que se ha accedido hace más de cuatro días (más de 96 horas).

```
root@client log]# find . -mtime -1
./lastlog
./wtmp
./tuned/tuned.log
./audit/audit.log
./secure
./sa/sa20
./boot.log
./cron
./maillog
./vboxadd-install-x11.log
./messages
./dmesg
```



Eche un vistazo a los criterios **-newer**, **-anewer** y **-cnewer**, que toman un archivo como parámetro. En este caso, find busca los archivos que son más recientes que el designado.

f. -perm

-perm permite efectuar búsquedas en las autorizaciones de acceso (derechos, SUID, SGID, Sticky). Se deben precisar los derechos en base 8 (valor octal) y completos. El carácter - colocado antes del valor octal significa que los archivos buscados deben tener al menos los permisos establecidos. En el ejemplo siguiente, se buscan los directorios sobre los cuales todo el mundo (user, group, others) tiene permiso de ejecución (permiso x, o sea 1).

```
[root@client log]# find -type d -perm -111
./tuned
./anaconda
./ntpstats
./pluto
./glusterfs
./chrony
./cups
```

g. -links y -inum

Aunque estos criterios se refieren a nociones más avanzadas del sistema de archivos, es buena idea presentarlos ahora. Podrá volver a ellos en cuanto el capítulo Los discos y el sistema de archivos le presente el funcionamiento interno de un sistema de archivos.

La opción **-links** permite una búsqueda por nombre de hard links. Puede precisar los signos + o - (más de n vínculos y menos de n vínculos). Un archivo normal único posee 1 vínculo. Un directorio, 2 vínculos (la entrada en el catálogo al cual pertenece y en el punto). Para una búsqueda de vínculos simbólicos, habrá que utilizar la opción **-type l**.

```
$ find . -type f -links +2 -print
./fic2
./hardlink3_fic2
./hardlink_fic2
./hardlink2_fic2
```

`-inum` permite una búsqueda por número de inodo. Es útil en el caso de una búsqueda de todos los vínculos que llevan un mismo número de inodo. El número de inodo puede verse mediante la opción `-i` del comando `ls`.

```
seb@slyserver:/var/log> ls -i
491891 acpid          491793 mail.info  491860 Xorg.0.log
491791 boot.log       491794 mail.warn  490686 Xorg.0.log.old
491729 boot.msg       492046 mcelog     492060 Xorg.1.log
seb@slyserver:/var/log> find . -inum 491791 -print
./boot.log
```

h. -regex e -iregex

`-regex` devolverá los valores correspondientes a las expresiones regulares proporcionadas. `-iregex` hace lo mismo sin discriminar entre mayúsculas y minúsculas.

```
[jolvares@client Documents]$ find . -regex '.*slyunix/t-shirt.*'
./slyunix/t-shirt
./slyunix/t-shirt/.xvpics
./slyunix/t-shirt/.xvpics/delante.jpg
./slyunix/t-shirt/.xvpics/detras.jpg
./slyunix/t-shirt/delante.jpg
./slyunix/t-shirt/delante.sxd
./slyunix/t-shirt/detras.jpg
./slyunix/t-shirt/detras.sxd
./slyunix/t-shirt/Thumbs.db
```

i. -depth, -maxdepth, -mindepth

Estos parámetros influyen en el orden y la profundidad de la búsqueda. `-depth` busca primero en los subdirectorios y después en el directorio en el que nos encontramos. `-maxdepth` va seguido de un número entero y solicita una recursividad hasta la profundidad de directorios indicada. `-mindepth` hace lo mismo pero para una profundidad mínima.

```
[seb@client ~]$ find / -maxdepth 2 -name bin
/usr/bin
/bin
[seb@client ~]$ find / -mindepth 2 -maxdepth 4 -name bin 2>/dev/null
/usr/bin
/usr/lib/debug/bin
/usr/share/locale/bin
/usr/local/bin
```

3. Comandos

Además de la opción `-print`, hay otras opciones que permiten efectuar una acción en los archivos encontrados. Los comandos son, de hecho, predicados como los anteriores, pero siempre arrojan el resultado verdadero. Se sitúan en el final de la línea y son, por lo tanto, siempre ejecutados. Aquí encontrará una lista no exhaustiva.

a. -ls

El criterio muestra información detallada en los archivos encontrados que corresponden al criterio, en lugar del simple nombre de archivo. La salida corresponde a un comando **ls** con los parámetros `-d`, `-i`, `-l` y `-s` (tamaño en bloques de 1 KB).

```
[jolivares@cliente Documentos]$ find -size +5000k -ls
101437675 5496 -rw-r--r-- 1 jolivares jolivares 5625432 feb 26 2004
./slyunix/logostif
101437683 5508 -rw-r--r-- 1 jolivares jolivares 5636322 mar 2 2004
./slyunix/pub_planeta.sxd
101437684 5508 -rw-r--r-- 1 jolivares jolivares 5636332 mar 2 2004
```

```
./slyunix/pub_planeta2.sxd
34386311 5392 -rw-r--r-- 1 jolivares jolivares 5521408 dic 11 2013
./slyunix/SITE_WWW/libro_eni/capitulo/capitulo_7.doc
968441 10992 -rw-r--r-- 1 jolivares jolivares 11254327 sep 13 2007
./slyunix/SITE_WWW/slyunix.sql jolivares
```

b. -exec

El criterio `-exec` va a ejecutar el comando colocado justo después con cada coincidencia encontrada. Es necesario hacer algunas observaciones:

- ▾ `-exec` debe ser obligatoriamente la última opción del comando **find**.
- ▾ El comando ejecutado por `-exec` debe terminarse con un «;». Este carácter especial debe escribirse con `\;` para que el shell no lo interprete.
- ▾ Con el fin de pasar como parámetro para el comando el archivo encontrado por find, hay que escribir `{}` (sustitución del archivo).

Ejemplo para borrar todos los archivos que terminan por «.mp3»:

```
$ find . -type f -name "*.mp3" -exec rm -f {} \;
```



El sistema no espera la finalización del comando **find** para ejecutar el comando **rm**. Lo ejecuta en cuanto **find** encuentra un archivo. Así, si el comando anterior le ha mostrado n archivos antes de que pensara interrumpirlo, estos n archivos ya han sido borrados.

c. -ok

El criterio `-ok` es idéntico a la opción `-exec`, pero, para cada coincidencia, se le requiere

una confirmación al usuario.

```
$ find . -inum 95 -ok rm -f {} \;  
< rm ... ./fic1 > (yes)? n  
< rm ... ./vínculo_fic1 > (yes)? y
```

4. Criterios AND / OR / NOT

Es posible combinar las opciones de criterio de selección. Si no se especifica ninguno, el Y lógico está implícito.

Criterio	Acción
-a, -and	AND, Y lógico, por defecto
-o, -or	OR, O lógico
!	Negación del criterio

Ejemplo con todos los archivos que no contienen fic en su nombre, y todos los archivos que no son ni normales ni directorios.

```
$ find . ! -name "*fic*" -print  
.  
./dir1  
./lista  
./mypass  
./users  
./lista2  
./ls.txt  
./pepito.tar.gz
```

```
./nohup.out
./lista_ls
./dir2
./seb1
./seb2
```

Listar de forma detallada los archivos diferentes de los archivos ordinarios o los directorios (enlaces simbólicos, archivos especiales).

```
$ find . ! \( -type f -o -type d \) -ls
  409  0 lrwxrwxrwx  1 oracle  system    4 Ago 14 15:21
./vínculo_fic1 -> fic1
  634  0 lrwxrwxrwx  1 oracle  system    4 Ago 14 15:21
./vínculo_fic2 -> fic2
```

5. Encontrar ejecutables

a. whereis

El comando **whereis** busca en las rutas de archivos binarios, del manual y de las fuentes los archivos que corresponden a los criterios facilitados.

```
$ whereis date
date: /bin/date /usr/share/man/man1/date.1.gz
/usr/share/man/man1p/date.1p.gz
```

Puede precisar algunos parámetros:

- ✓ **-b** únicamente para los binarios,
- ✓ **-m** únicamente para los manuales,
- ✓ **-s** únicamente para las fuentes.
- ✓ **-B** (y después **-M** y **-S**): modifica la lista de los directorios de búsqueda para

cada uno de los tipos.

Se buscan los archivos por defecto en:

```
{bin,sbin,etc}
/usr/{lib,bin,old,new,local,games,include,etc,src,man,sbin,X386,TeX,
g++-include}
/usr/local/{X386,TeX,X11,include,lib,man,etc,bin,games,emacs}
```

Por lo tanto, no se sorprenda de obtener lo siguiente incluso si el archivo `/etc/passwd` no es un ejecutable, el comando se basa en las rutas por tipo y no en el mismo tipo:

```
$ whereis -b passwd
passwd: /usr/bin/passwd /etc/passwd /etc/passwd.old
/etc/passwd.YaST2save /etc/passwd.vipwKSnTgH /usr/bin/X11/passwd
```

b. which

El comando **which** busca un comando en el PATH (ruta de los ejecutables) y le facilita el primero que encuentra:

```
$ which date
/bin/date
```

En Linux hay comandos con el mismo nombre en varias rutas. Debe precisar el parámetro `-a` para que **which** siga su búsqueda. Sin embargo, si usted ejecuta el comando sin especificar una de esas rutas, se ejecutará el primero por defecto.

```
$ which -a passwd
/usr/bin/passwd
/usr/bin/X11/passwd
```

c. locate

El comando **locate** busca un archivo según el modelo dado en una base de datos de archivos construida por el comando **updatedb**.

El comando **updatedb** recorre una serie de rutas en las que ejecuta un **find** y almacena todos los resultados en una base indexada. Esto evita, por lo tanto, efectuar de nuevo un **find** para las búsquedas clásicas. En la práctica, basta con pasarle a **updatedb** la lista de las rutas o no incluir los archivos en la base indexada.

contrab suele lanzar **updatedb** a diario. A veces se colocan los parámetros del comando en un archivo `/etc/sysconfig/locate` o en `/etc/updatedb.conf`.


```
$ cat /etc/updatedb.conf
PRUNE_BIND_MOUNTS = "yes"
PRUNEFS = "9p afs anon_inodefs auto autofs bdev binfmt_misc cgroup cifs
coda configfs cpuset debugfs devpts ecryptfs exofs fuse fuse.sshfs fusectl
gfs gfs2 gpfs hugetlbfs inotifyfs iso9660 jffs2 lustre mqueue ncpfs nfs
fs4 nfsd pipefs proc ramfs rootfs rpc_pipefs securityfs selinuxfs sfs sockfs
sysfs tmpfs ubifs udf usbfs ceph fuse.ceph"
PRUNENAMES = ".git .hg .svn .bzip .arch-ids {arch} CVS"
PRUNEPATHS = "/afs /media /mnt /net /sfs /tmp /udev /var/cache/ccache /var/
lib/yum/yumdb /var/lib/dnf/yumdb /var/spool/cups /var/spool/squid /var/tmp
/var/lib/ceph"
```

En este caso, se ejecuta el comando siguiente:

```
# updatedb -f 9p,afs,anon_inodefs,... -e /afs,/media,/mnt,...
```

Si usted ejecuta el comando de esta manera, puede que éste consuma todos los recursos de procesador de su máquina. Por eso, se ejecuta **updatedb** con una prioridad baja.

```
$ locate passwd
/etc/passwd
/etc/passwd-
/etc/passwdqc.conf
/etc/pam.d/passwd
/etc/security/opasswd
```



```
/usr/bin/gpasswd  
/usr/bin/grub2-mkpasswd-pbkdf2  
/usr/bin/mkpasswd  
/usr/bin/passwd  
/usr/lib/firewalld/services/kpasswd.xml  
...
```