

FIND

NOMBRE

find - busca ficheros en un árbol de directorios

SINOPSIS

find

[camino...] [expresión]

DESCRIPCIÓN

Esta página del Manual documenta la versión de GNU de **find**. **find** recorre el árbol de directorio cuya raíz reside en cada nombre de fichero dado, evaluando de izquierda a derecha la expresión especificada, según las reglas de precedencia (vea la sección de OPERADORES), hasta que se conoce el resultado (la parte izquierda es falsa para operaciones *and*, verdadera para *or*), en cuyo punto **find** se mueve al siguiente nombre de fichero.

El primer argumento que comience con ``-'`, ``('`, ``)'`, ``,'`, o ``!'` se toma como el principio de la expresión; cualesquier argumentos antes de él son caminos a recorrer, y cualesquier argumentos tras él forman el resto de la expresión. Si no se da ningún camino, se toma el directorio de trabajo. Si no se da ninguna expresión, se sobreentiende la expresión ``-print'`.

find acaba con código de estado 0 si todos los ficheros se procesan satisfactoriamente, mayor que 0 si ocurre un error.

EXPRESIONES

La expresión está hecha de opciones (que afectan a todas las operaciones en vez de al procesamiento de un fichero específico, y siempre devuelven verdad), comparaciones (que devuelven un valor verdadero o falso), y acciones (que tienen efectos colaterales y devuelven un valor verdadero o falso), todos separados por operadores. Donde se omita el operador, se sobreentiende `-and`. Si la expresión no contiene ninguna acción, aparte de `-prune`, se ejecuta `-print` en todos los ficheros para los que la expresión dé verdadera.

OPCIONES

Todas las opciones siempre devuelven verdadero. Siempre tienen efecto, en lugar de ser procesadas solamente cuando se alcance su sitio en la expresión. Por lo tanto, por claridad, lo mejor es ponerlas siempre al principio de la expresión.

`-daystart`

Mide los tiempos (para `-amin`, `-atime`, `-cmin`, `-ctime`, `-mmin`, y `-mtime`) desde el principio del día de hoy en lugar de desde hace 24 horas.

`-depth`

Procesa cada contenido del directorio antes del propio directorio.

`-follow`

Desreferencia enlaces simbólicos. Implica `-noleaf`.

`-help`, `--help`

Muestra un sumario del modo de empleo de **find** y acaba.

-maxdepth *niveles*

Desciende como mucho *niveles* (un entero no negativo) niveles de directorios por debajo de los argumentos de la línea de órdenes. '-maxdepth 0' significa que sólo se aplicarán las comparaciones y acciones a los argumentos de la línea de órdenes.

-mindepth *niveles*

No aplica comparaciones ni acciones a niveles menores de *niveles* (un entero no negativo). '-mindepth 1' significa procesar todos los ficheros excepto los argumentos de la línea de órdenes.

-mount

No descender a directorios en otros sistemas de ficheros. Un nombre alternativo para -xdev, por compatibilidad con algunas otras versiones de **find**.

-noleaf

No optimiza suponiendo que los directorios contienen dos subdirectorios menos que lo que dice su número de enlaces duros. Esta opción es necesaria cuando se recorren sistemas de ficheros que no siguen el convenio de Unix sobre los enlaces de directorios, como sistemas de ficheros ISO-9660 (de un CD-ROM) o FAT (de MS-DOS o Windows), o puntos de montaje de volúmenes AFS (Andrew File System). Cada directorio de un sistema de ficheros normal de Unix (UFS, ext2fs, etc.) tiene por lo menos dos enlaces duros: su nombre y su entrada '.' (punto). Además, sus subdirectorios (si los hay) tienen cada uno una entrada '..' enlazada a este directorio. Cuando **find** está examinando un directorio, después de que ha hecho stat() en dos subdirectorios menos de lo que dice el número de enlaces del directorio, sabe que el resto de las entradas del directorio son no-directorios (ficheros 'hoja' ['leaf' en inglés] en el árbol del directorio). Si sólo hay que examinar los nombres de ficheros, no hay necesidad de llamar a stat() sobre ellos, lo cual proporciona un incremento significativo en la velocidad del recorrido.

-version, --version

Muestra el número de versión de **find** y acaba.

-xdev

No descende a directorios en otros sistemas de ficheros. Vea -mount más arriba.

COMPARACIONES

Los argumentos numéricos pueden especificarse como

+*n*

para mayor que *n*,

-*n*

para menor que *n*,

n

para exactamente *n*.

-amin *n*

Se ha accedido al fichero por última vez hace *n* minutos.

-anewer *fichero*

Se ha accedido al fichero por última vez más recientemente que cuando *fichero* fue modificado. -anewer está afectado por -follow solamente si -follow va antes de -anewer en la línea de órdenes.

-atime *n*

Se ha accedido al fichero por última vez hace *n*×24 horas.

-cmin *n*

El status del fichero se cambió por última vez hace *n* minutos.

-cnewer *fichero*

El status del fichero se cambió por última vez más recientemente que cuando se modificó *fichero*. -cnewer está afectado por -follow solamente si -follow va antes de -cnewer en la línea de órdenes.

-ctime *n*

El status del fichero se cambió por última vez hace $n \times 24$ horas.

-empty

El fichero está vacío y es o un fichero regular o un directorio.

-false

Siempre da falso.

-fstype *tipo*

El fichero está en un sistema de ficheros de tipo *tipo*. Los tipos válidos de sistemas de ficheros varían según la versión de Unix; una lista incompleta de tipos de sistemas de ficheros que se aceptan en algunas versiones de Unix es: ufs, 4.2, 4.3, nfs, tmp, mfs, S51K, S52K. Se puede usar -printf con la directiva %F para ver los tipos de sus sistemas de ficheros. (En Linux, cat /proc/filesystems dará una lista de los tipos de sistemas de ficheros reconocidos.)

-gid *n*

El GID numérico del fichero es *n*.

-group *grupo*

El fichero pertenece al grupo *grupo* (se permite un GID numérico).

-lname *patrón*

Como -lname, pero la concordancia no hace caso de mayúsculas ni minúsculas.

-iname *patrón*

Como -name, pero la concordancia no hace caso de mayúsculas ni minúsculas. Por ejemplo, los patrones ``fo*'` y ``F??'` concuerdan con los nombres de ficheros ``Foo'`, ``FOO'`, ``foo'`, ``fOo'`, etc.

-inum *n*

El fichero tiene de número-i *n*.

-ipath *patrón*

Como -path, pero la concordancia no hace caso de mayúsculas ni minúsculas.

-iregex *patrón*

Como -regex, pero la concordancia no hace caso de mayúsculas ni minúsculas.

-links *n*

El fichero tiene *n* enlaces.

-lname *patrón*

El fichero es un enlace simbólico cuyos contenidos concuerdan con el patrón del shell *patrón*. Los metacaracteres no tratan especialmente a ``'` ni a ``.``.

-mmin *n*

Los datos del fichero fueron modificados por última vez hace *n* minutos.

-mtime *n*

Los datos del fichero fueron modificados por última vez hace $n \times 24$ horas.

-name *patrón*

La base del nombre del fichero (el camino sin los directorios) concuerda con el patrón del shell *patrón*. Los metacaracteres (``*'`, ``?'` y ``[]'`) no concuerdan con un ``.`` al principio del nombre base. Para omitir un directorio y los ficheros bajo él, emplee -prune; vea un ejemplo en la descripción de -path.

-newer *fichero*

El fichero fue modificado más recientemente que *fichero*. -newer es afectado por -follow solamente si -follow va antes de -newer en la línea de órdenes.

-nouser

Ningún usuario se corresponde con el UID numérico del fichero.

-nogroup

Ningún grupo se corresponde con el GID numérico del fichero.

-path *patrón*

El nombre del fichero concuerda con el patrón del shell *patrón*. Los metacaracteres no tratan de forma especial a ``'` o a ``.'`; así, por ejemplo,

`find . -path './sr*sc'`

mostrará una entrada para un directorio llamado `./src/misc` (si es que existe). Para omitir un árbol entero de directorio, emplee `-prune` en vez de comprobar cada fichero en el árbol. Por ejemplo, para pasar el directorio `src/emacs` y todos los ficheros y directorios bajo él, y mostrar los nombres de los otros ficheros encontrados, haga algo como esto:

`find . -path './src/emacs' -prune -o -print`

-perm *modo*

Los bits de permiso del fichero son exactamente *modo* (octal o simbólico). Los modos simbólicos utilizarán 0 como punto de partida.

-perm *-modo*

Todos los bits de permiso *modo* están activos para el fichero.

-perm *+modo*

Cualquiera de los bits de permiso de *modo* está activo para el fichero.

-regex *patrón*

El nombre del fichero concuerda con la expresión regular *patrón*. Esta concordancia es en el camino entero, no es una búsqueda. Por ejemplo, para un fichero llamado `./fubar3`, Ud. puede emplear la expresión regular ``.bar.'` or ``.b.*3'`, mas no ``b.*r3'`.

-size *n*[bckw]

El fichero emplea *n* unidades de espacio. Las unidades son bloques de 512 bytes si no se dice otra cosa o si la letra ``b'` sigue a *n*, bloques de bytes si se pone la letra ``c'`, kilobytes con la ``k'`, o palabras de 2 bytes con la ``w'`. El tamaño no cuenta bloques indirectos, pero sí cuenta bloques en ficheros esparcidos que no están realmente asignados.

-true

Siempre da verdadero.

-type *c*

El fichero es de tipo *c*:

b	especial de bloques (tamponado)
c	especial de caracteres (sin tamponar)
d	directorio
p	tubería con nombre (FIFO)
f	fichero regular
l	enlace simbólico
s	zócalo (socket)

-uid *n*

El UID numérico del propietario del fichero es *n*.

-used *n*

Se accedió por última vez al fichero *n* días después de que se cambió por última vez su estado.

-user *usuario*

El propietario del fichero es el usuario *usuario* (se permite un UID numérico).

-xtype *c*

Lo mismo que `-type` a menos que el fichero sea un enlace simbólico. Para enlaces simbólicos: si no se ha dado `-follow`, verdad si el fichero es un enlace a un fichero de tipo *c*; si no, verdad

si *c* es ``l'`. En otras palabras, para enlaces simbólicos, `-xtype` comprueba el tipo del fichero que `-type` no comprueba.

ACCIONES

`-exec orden ;`

Ejecuta *orden*; verdad si se devuelve el status 0. Todos los argumentos que siguen a **find** se toman como de la orden hasta que se encuentre uno consistente en ``;'`. La cadena ``{'` se reemplaza por el nombre del fichero que se está procesando; en cualquier sitio de los argumentos de la orden, no sólo en argumentos donde esté sola, como en algunas otras versiones de **find**. Ambas construcciones pueden necesitar 'escaparse' (con una ``\'`) o entrecorillarse, para protegerlas de la expansión que efectuaría el shell. La orden se ejecuta en el directorio de comienzo.

`-fls fichero`

Verdad; como `-ls` pero escribe en *fichero* como `-fprint`.

`-fprint fichero`

Verdad; escribe el nombre completo del fichero en *fichero*. Si *fichero* no existe cuando se ejecuta **find**, se crea; si existe, se trunca. Los nombres de ficheros ```/dev/stdout"` y ```/dev/stderr"` se manejan de manera especial; se refieren a la salida estándar y la salida estándar de errores, respectivamente.

`-fprint0 fichero`

Verdad; como `-print0` pero escribe en *fichero* como `-fprint`.

`-fprintf fichero formato`

Verdad; como `-printf` pero escribe en *fichero* como `-fprint`.

`-ok orden ;`

Como `-exec` pero pregunta primero al usuario (en la entrada estándar); si la respuesta no comienza con ``y'` o ``Y'` (o quizás el equivalente local, en español ``s'` o ``S'`), no se ejecuta la orden, y devuelve falso.

`-print`

Verdad; imprime el nombre completo del fichero en la salida estándar, seguido por un salto de línea.

`-print0`

Verdad; imprime el nombre completo del fichero en la salida estándar, seguido por un carácter nulo. Esto permite que nombres de ficheros que contengan saltos de línea sean correctamente interpretados por programas que procesen la salida de **find**.

`-printf formato`

Verdad; imprime *formato* en la salida estándar, interpretando secuencias de escape ``\'` y directivas ``%'`. Las anchuras de campo y precisiones pueden especificarse como en la función de C ``printf'`. De forma contraria a lo que hace `-print`, `-printf` no añade un salto de línea al final de la cadena. Las secuencias de escape y directivas son:

``a`

La campana de alarma.

``b`

Espacio atrás.

``c`

Para la impresión de este formato inmediatamente y vuelca el búfer de salida.

``f`

Salto de página.

``n`

Salto de línea.

<code>\r</code>	Retorno de carro.
<code>\t</code>	Tabulador horizontal.
<code>\v</code>	Tabulador vertical.
<code>\\</code>	Una barra inclinada inversa literal (<code>\</code>).

Un carácter ``\`` seguido por otro cualquiera distinto de los anteriores se trata como un carácter ordinario, por lo que se imprimirán ambos.

<code>%%</code>	Un signo de porcentaje literal <code>`%`</code> .
<code>%a</code>	El tiempo de último acceso al fichero en el formato devuelto por la función de C <code>`ctime()`</code> .
<code>%Ak</code>	El tiempo de último acceso al fichero en el formato especificado por <i>k</i> , que es o bien <code>`@`</code> o bien una directiva válida para la función de C <code>`strftime()`</code> . Los valores posibles para <i>k</i> se listan abajo; algunos de ellos pueden no estar disponibles en todos los sistemas, debido a diferencias en la función <code>`strftime()`</code> entre sistemas.
<code>@</code>	segundos desde el 1 de Enero de 1970 a las 00:00 h GMT.

Campos de tiempo:

<code>H</code>	hora (00..23)
<code>I</code>	hora (01..12)
<code>k</code>	hora (0..23)
<code>l</code>	hora (1..12)
<code>M</code>	minuto (00..59)
<code>p</code>	AM o PM, en su representación local
<code>r</code>	hora, formato de 12 horas (hh:mm:ss [AP]M)
<code>S</code>	segundo (00..61)
<code>T</code>	tiempo, formato de 24 horas (hh:mm:ss)
<code>X</code>	representación local de la hora (H:M:S)
<code>Z</code>	zona horaria (e.g., CET), o nada si el huso horario no es determinable

Campos de fecha:

a	nombre local abreviado del día de la semana (Dom..Sáb)
A	nombre local completo del día de la semana, longitud variable (Domingo..Sábado)
b	nombre local abreviado del mes (Ene..Dic)
B	nombre local completo del mes, longitud variable (Enero..Diciembre)
c	fecha y hora en representación local (Sáb 4 Nov 12:02:33 CET 1989)
d	día del mes (01..31)
D	fecha (mm/dd/aa)
h	lo mismo que b
j	día del año (juliano) (001..366)
m	mes numérico (01..12)
U	número de la semana del año con el Domingo como primer día de la semana (00..53)
w	día de la semana, numérico (0..6)
W	número de la semana del año como el Lunes como el primer día de la semana (00..53)
x	representación local de la fecha (dd/mm/aa)
y	los últimos dos dígitos del año (00..99)
Y	año (1970...)
%b	El tamaño del fichero en bloques de 512 bytes (redondeado)
%c	El tiempo del último cambio del estado del fichero en el formato devuelto por la función de C <code>`ctime()</code> '.
%Ck	El tiempo del último cambio del estado del fichero en el formato especificado por <i>k</i> , que es el mismo que para %A.
%d	La profundidad en el árbol del directorio; 0 significa que el fichero es un argumento de la línea de órdenes.
%f	El nombre del fichero sin los directorios (sólo el último elemento del camino).
%F	Tipo del sistema de ficheros donde está el fichero; este valor puede emplearse para <code>-fstype</code> .

%g	El nombre de grupo al que pertenece el fichero, o el GID numérico si el grupo no tiene nombre.
%G	El GID numérico del grupo al que pertenece el fichero.
%h	Los directorios del camino del fichero (todos los componentes del camino menos el último elemento).
%H	Argumento de la línea de órdenes bajo el que se ha encontrado el fichero.
%i	El número-i del fichero (en base 10).
%k	El tamaño del fichero en bloques de un kB (redondeado).
%l	El objeto de un enlace simbólico (la cadena vacía si el fichero no es un enlace simbólico).
%m	Los bits de permiso del fichero (en octal).
%n	El número de enlaces duros del fichero.
%p	El nombre del fichero.
%P	El nombre del fichero con el nombre del argumento de la línea de órdenes bajo el que se ha encontrado, quitado.
%s	El tamaño del fichero en bytes.
%t	El tiempo de última modificación del fichero en el formato devuelto por la función de C <code>`ctime()`</code> .
%Tk	El tiempo de última modificación del fichero en el formato especificado por <i>k</i> , que es el mismo que para %A.
%u	El nombre del usuario propietario del fichero, o el UID si el usuario no tiene nombre.
%U	El UID numérico del propietario del fichero.

Un carácter ``%'` seguido por cualquier otro distinto de los recién mencionados no es tenido en cuenta (pero el otro carácter se imprime).

-prune

Si `-depth` no se da, devuelve verdad; no desciende por el directorio en curso.
Si `-depth` se da, devuelve falso; no tiene otro efecto.

-ls

Verdad; lista el fichero actual en el formato de ``ls -dils'` en la salida estándar. Los números de bloques son en bloques de 1 kB, a menos que la variable de ambiente `POSIXLY_CORRECT` esté definida, en cuyo caso los bloques son de 512 B.

OPERADORES

Listados en orden de precedencia decreciente:

(*expr*)

Fuerza la precedencia.

! *expr*

Verdad si *expr* es falsa.

-not *expr*

Lo mismo que ! *expr*.

expr1 expr2

Y (implícito); *expr2* no se evalúa si *expr1* es falsa.

expr1 -a *expr2*

Lo mismo que *expr1 expr2*.

expr1 -and *expr2*

Lo mismo que *expr1 expr2*.

expr1 -o *expr2*

O; *expr2* no se evalúa si *expr1* es verdad.

expr1 -or *expr2*

Lo mismo que *expr1* -o *expr2*.

expr1 , *expr2*

Lista; *expr1* y *expr2* son, ambos, evaluados siempre. El valor de *expr1* se descarta; el valor de la lista es el de *expr2*.