find

opciones

- -name : busca ficheros por nombre
- -iname : busca ficheros por nombre sin tener en cuenta mayúsculas o minúsculas
- -path : busca en el path completo (directorio+nombre)
- -ipath : busca en el path completo (directorio+nombre) no Case Sensitive
- -maxdepth <n> : número máximo de directorios que profundiza, 1 se queda en el actual
- -mindepth <n> : número de directorios « de profundidad» a los que empieza a mirar
- -mtime -<n*24h> : número de 24h atrás en los que el fichero ha sufrido algún cambio
- -mmin -<minutos> : número de minutos en los que el fichero ha sufrido algún cambio
- -atime -<n*24h> : número de 24h atrás en los que el fichero ha sido accedido
- -amin -<minutos> : número de minutos en los que el fichero ha sido accedido
- -ctime -<n*24h> : número de 24h atrás en los que el inodo del fichero/directorio ha sido modificado
- -cmin -<minutos> : número de minutos en los que el inodo del fichero/directorio ha sido modificado
- -newer <fichero> : buscar ficheros más recientes que un fichero determinado
    - `$ touch -d «6 May 2012 18:12:11» marca_tiempo ; find . -newer marca_tiempo`
- -cnewer
- -anewer
- -atime
- -daystart : indica que el día empieza a las 00:00 y no 24h antes
    - `$ find . -daystart -atime 1 # busca un fichero modificado hoy, pero no 24h antes`
- -size <tamaño><unidad> : busca por tamaño de archivo
    - c : bytes
    - w : 2 bytes word
    - k : kilobytes
    - b : bloque de 512 bytes
    - `$ find . -size 100c -and -size 200c # busca ficheros entre 100 y 200 bytes`
- -empty : busca ficheros vacíos
    - más eficiente que -size 0c
- booleanos
    - -not : ! en formato abreviado, escapado en bash
    - -and : -a en formato abreviado
    - -or : -o en formato abreviado
    - (...) : los paréntesis van escapados
    - ,
- -type : buscar en tipos específicos
    - d : directorios
    - f : ficheros
    - l : enlaces simbólicos
- -fprint <file> : exporta a <file>
    - es más eficiente que usar redirectores > o »
- -printf <formato><tar,\\n>
    - %p : nombre de fichero, incluidos directorios
    - %m : permisos del fichero
    - %f : nombre del fichero, sin directorio
    - %g : nombre del grupo al que pertenece el fichero
    - %h : muestra el nombre del directorio sin el nombre del fichero
    - %u : nombre del usuario al que pertenece el fichero
- -prune : convierte la expresión precedente en «no quiero esto»
- -user <user> : el fichero pertenece al usuario <user>

- -nouser : el fichero tiene un propietario que no está en /etc/passwd
- -uid <uid>
- -group <grupo>
- -nogroup : el fichero tiene un grupo que no está en /etc/groups
- -gid <gid>
- -perm XXX : busca por permisos
    - -perm -g=r → busca que en el grupo tenga permisos de lectura
- -exec <acción> {}\;
    - ejecuta la acción sobre el resultado del find (se sustituye cada entrada en {}
    - es más eficiente que «pipear» el resultado |
- -ok : lo mismo que -exec, pero con confirmación por cada resultado de find
- -regexp <regexp>
- -iregexp
- -fstype
    - msdos
- xdev: solo busca en el filesystem actual, si hay otros montados como directorios en subdirectorios no se buscará en ellos

ejemplos

básicos

- buscar ficheros que empiecen por cadena y algo más: `find . -name cadena\*`
- buscar todos los fichero excepto los que sean en minúsculas: `find . -iname cadena\* -not -name cadena\*`
- buscar en directorios+carpetas, mostrar solo ficheros : `find . -path \*cadena -type f`
- buscar 2 juegos de ficheros diferentes, vuelca el resultado en ficheros: `find . -type f \( -name \*.php -fprint php_files , -name \*.js -fprint js_files \)`
- buscar unos ficheros excepto algunos: `find \( -path <criterio_no_quiero#1> -o -path <criterio_no_quiero#2> \) -prune -o -path <criterio_quiero>`
- borrar ficheros siguiendo un criterio: `find -iname \*.mp3 -exec rm {} \;`
- ejecutar programa en ficheros encontrados: `find -iname «MyCProgram.c» -exec md5sum {} \;`
- mostrar los ficheros que contienen una cadena: '' find . -exec grep -l «cadena» {} \;''
- hacer copias de seguridad de ciertos ficheros: `find -name «*.txt» cp {} {}.bkup \;`
- renombrar ficheros: `find -name «*.txt» -exec mv {} `basename {} .htm`.html \;`
- cambiar espacios por subrayados: `find . -type f -iname “*.mp3″ -exec rename “s/ /_/g” {} \;`
- buscar los 5 archivos más grandes: `find . -type f -exec ls -s {} \; | sort -n -r | head -5`
- copiar archivos de extensiones diferentes en una carpeta:
    - `find . -type f -iname \*.jpg -o -iname\*.docx | xargs cp -t /path/destino/`
    - `find . -type f -regex '.*\.\(jpg\|docx\)' | xargs cp -t /path/destino`
- cuenta directorios: `find <PATH> -maxdepth 1 -type d -printf '.' | wc -c`
- Borrar recursivamente directorios vacios: `find . -type d -empty -delete`
- Borrar recursivamente directorios vacios: `find . -type d -empty -print0 | xargs -0 rmdir`
- Borrar fichero por inodo: `ls -li; find . -inum <número inodo> | xargs rm`
- Buscar una cadena en una serie de ficheros: `find . -name «*.java» -print0 | xargs -0 grep -i «.*Legacy.*xmi»`
- buscar en el directorio <dir> la cadena <cadena>: `find <dir> -name «<cadena>»`
- borrar ficheros que cumplan un requisito: `find . -type f -name '*.class' -exec rm -vf {} \;`
- buscar archivos a partir de un determinado tamaño: `find / -type f -size +20000k -exec ls -lh {} \; 2> /dev/null | awk '{ print $NF «: » $5 }' | sort -nrk 2,2`
- localizar archivos del tipo imagen o video y borrarlos: `find . -type f -exec file {} \; | awk -`

```
F: ‘{if ($2 ~/image|video/) print $1 }’ | xargs rm -f
```

tiempo

- buscar un fichero modificado hoy/ayer:
  - hoy: `find . -daystart -mtime 0`
  - ayer: `$ find . -daystart -mtime 1`
- buscar ficheros más recientes que otro de referencia: `find . -name '*.java' -newer build.xml -print`
- borrar ficheros de más de 30 días: `find /path/ -type f -mtime +30 -exec rm -f {} \;`

```
find ${BACKUP_PATH} -daystart -type f -mtime +31 -not -name "*01.sql.tgz" -not -name "*01.sql.gz" -not -name "enchive.*" -exec echo rm {} \;
find ${BACKUP_PATH} -daystart -type f -mtime +31 -not -name "*01.sql.tgz" -not -name "*01.sql.gz" -not -name "enchive.*" -delete
```

- localizar ficheros de menos de 30 días: `find . -type f -mtime -30`
- localizar ficheros entre dos fechas: `find -type f -name «*.dcm» -newermt 2016-04-08 ! -newermt 2016-04-09`
- buscar ficheros más recientes que otro fichero de referencia: `find . -type f -newer <fichero-referencia>`
  - si queremos que sean anteriores al fichero de referencia: `find . -type f ! -newer <fichero-referencia>`
- buscar solo en archivos «ocultos» (en este caso 15 minutos atrás): `find . -mmin -15 \( ! -regex ".*/\..*" \)`
- buscar entre últimos días excluyendo un patrón por nombre (por refinar):`find . -type f -mtime +7 -not -regex «.*/[2][0-9][0-1][0-9][0-1][1,6].*»`
  - la idea es que excluya los XXXX01 y XXXX16, esto actualmente hace match en 01,06,11,16
  - https://newbedev.com/how-to-use-regex-with-find-command
  - https://unix.stackexchange.com/questions/70933/regular-expression-for-finding-double-characters-in-bash
  - https://stackoverflow.com/questions/6844785/how-to-use-regex-with-find-command
  - https://regex101.com/
  - https://towardsdatascience.com/regular-expressions-clearly-explained-with-examples-822d76b037b4

scripting

- ejecutar varias instrucciones sobre los items localizados:

```
find ... | while read -r file; do
    echo "look at my $file, my $file is amazing";
done
```

- preservar 1 mes de logs, guardar 1 de los anteriores:

```
#!/bin/bash
PRESERVE=$(/bin/date +'%Y-%m-01' -d '-1 month')
find . -daystart -type f -mtime +31 -not -mtime +60 -name "*.log" -not -name "*${PRESERVE}*" -exec rm {} \;
```

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/linux:bash:find

Last update: **02/09/2024 11:10**