El comando **find** de Linux es extremadamente potente, esto es, si logras usarlo adecuadamente. No hay nada mejor para hacer todo tipo de búsquedas de archivos y carpetas que este comando.

Hay por supuesto otros comandos de búsqueda como **awk** , **sed** y **grep** pero están más enfocados a buscar "dentro" de los archivos. **find** es mucho más útil para encontrar archivos y directorios. En este artículo aprenderás a usar **find** como todo un experto y después puedas aplicarlo en usos administrativos de todo tipo.

Sintaxis de find

La sintaxis es muy simple:

Código:
find [ruta] [expresión_de_búsqueda] [acción]

La [ruta] es cualquier directorio o path que se quiera indicar y desde donde inicia la búsqueda, ejemplos pueden ser " */etc* ", " */home/usuario* ", *"/", "."* si no se indica una ruta se toma en cuenta entonces el directorio donde se este actualmente, es decir el directorio de trabajo actual, que es lo mismo que indicar punto ".". De hecho es posible indicar más de un directorio de búsqueda como se verá más adelante en un ejemplo.

La [expresión_de_búsqueda] es una o más opciones que puede devolver la búsqueda a realizar en si o acciones a realizar sobre la búsqueda, si no se indica ninguna expresión de búsqueda se aplica por defecto la opción *-print* que muestra el resultado de la búsqueda.

La [acción] es cualquier comando de Linux invocado a ejecutarse sobre cada archivo o directorio encontrado con la [expresión_de_búsqueda].

Los tres argumentos anterior son enteramente opcionales

Búsquedas básicas

El siguiente ejemplo busca todos los archivos que contengan en su nombre "reporte" desde la raíz:

Código:
find / -name reporte

```
find / -iname Reporte  (lo mismo, pero sin tomar en cuenta mayúsculas y minúsculas)
```

El uso de expresiones regulares en lo que se busca es válido:

Código:

```
find / -name "[0-9]*"     (todo lo que empieze con un dígito)
find / -name "[Mm]*"      (todo lo que empieze con un la letra M o m)
find / -name "[a-m]*.txt" (todo lo que empieze entre a y m y termine en ".txt")
```

Busca bajo /home todos los archivos que pertenezcan al usuario mario

Código:

```
find /home -user mario

(lo mismo y que contengan con "enero" como en reporte_enero2011)
find /home -user mario -name "*enero*"
```

No estás limitado a un solo directorio, indica más de uno a buscar antes de las expresiones:

Código:

```
find /etc /usr /var -group admin

(busca en tres directorios todos los archivos o
subdirectorios que pertenezcan al grupo 'admin')
```

Búsquedas a través del tiempo

Varias opciones aceptan argumentos numéricos, estos pueden ser indicados de tres maneras

posibles:

Código:
+n busca valores mayor que n -n busca valores menor que n n busca exactamente el valor n

Buscar todos los archivos que hayan cambiado en los últimos 30 minutos:

Código:
find / -mmin -30 -type f los modificados exactamente hace 30 minutos: find / -mmin 30 -type f

O si deseas buscar en un rango específico de minutos, con este ejemplo buscarías todos los directorios que hayan cambiado hace más de 10 minutos (+10) y menos de 30 (-30)

Código:
find / -mmin +10 -mmin -30 -type d aunque lo anterior sería mas exacto decir los modificados hace 11 minutos o más y 29 minutos o menos, ya que como se vio anteriormente +n y -n indican "mayor que" y "menor que", el ejemplo correcto sería entonces: find / -mmin +9 -mmin -31 -type d

find ofrece varias opciones de búsqueda por tiempo, pero las principales son: *-amin, -atime, -cmin, ctime, -mmin y -mtime* . "min" es para periodos de minutos y "time" para periodos de 24 horas.

Los que empiezan con "a" (access) indica el tiempo en que fue accedido (leido) por última vez un archivo. Los que empiezan con "c" (change) indica el tiempo que cambió por última vez el

status de un archivo, por ejemplo sus permisos. Los que empiezan con "m" (modify) indica el tiempo en que fue modificado (escrito) por última vez un archivo.

Una consideración a tener con las búsquedas *-atime, -ctime y -mtime* es que el tiempo se mide en periodos de 24 horas y estos son siempre truncados, con ejemplos es más claro:

Código:
find . -mtime 0 (busca archivos modificados entre ahora y hace un dia) find . -mtime -1 (busca archivos modificados hace menos de un dia) find . -atime 1 (busca archivos accedidos entre hace 24 y 48 horas) find . -ctime +1 (busca archivos cuyo status haya cambiado hace más de 48 horas)

Comparaciones con -and, -or y -not

find también incluye operadores booleanos que la hace una herramienta aun más últil:

Código:
find /home -name 'ventas*' -and -mmin 120 find /home -name 'reporte[_-]*' -not -user sergio find /home -iname '*enero*' -or -group gerentes

El primer ejemplo busca todos los archivos que comiencen con 'ventas' Y que hayan sido modificados o cambiados en las últims dos horas (120 minutos).

El segundo ejemplo busca todos los archivos que comiencen con 'reporte' y después siga un _ o un - y que NO pertenezcan al usuario sergio.

El tercer ejemplo busca todos los archivos que contengan la palabra enero, Enero, ENERO, etc. (sin importar si lleva mayúsculas o minúsculas) O cualquier otro archivo que encuentre que pertenezca al grupo 'gerentes'.

Estas opciones de booleanos tienen su correspondiente abreviatura:
- *and* se puede indicar también como - *a*
- *or* se puede indicar también como - *o*
- *not* se puede indicar también como *!*

El tamaño si importa

Una de las actividades básicas de un administrador de sistemas Linux es monitorizar el tamaño de archivos, sobre todo de usuarios. Con **find** es muy fácil realizar búsquedas por tamaño, se indica con la opción *-size* , se aplican las mismas reglas para argumentos númericos (+n -n n).

Código:
find /var/log -size +15000k -name "*.jpg" (busca archivos mayores a 15 megas del tipo jpg) find $HOME -800c (busca en tu home todos los archivos menores a 800 bytes (799 realmente)) (archivos de tamaño comprendidos entre 1mb y 10mb) find . -size +1000k -and -size -10000k

Se admiten cuatro parámetros después del número en *-size* :

c = bytes
w = 2 byte words
k = kilobytes
b = 512-byte bloques

Para buscar archivos vacios puedes entonces hacer lo siguiente:

Código:
find . -size 0c (Aunque la opción -empty hace lo mismo más eficientemente) find . -empty

Cualquiera de los ejemplos anteriores dará un aburrido listado de los archivos y sus rutas. Si lo que quieres es realizar una acción (ejecutar un comando) sobre estos usa entonces la opción entonces *-exec* .

A escena -exec, el poder aumenta

- *exec* permite ejecutar acciones sobre el resultado de cada línea o archivo devuelto por **find** , o en otras palabras permite incorporar comandos externos para ejecutar sobre cada resultado devuelto. Muy interesante. Asi por ejemplo, si queremos buscar todos los archivos mayores a 3 megas en /var y además mostrar su salida en formato **ls** , podemos hacer lo siguiente:

Código:
find /var -size +3000k -exec ls -lh {} \;

Después de **ls -lh** que nos devuelve una salida formateada de **ls** se indica la cadena '{}' que se sustituye por cada salida de **find** .

No hay límite para lo que se puede lograr, asi por ejemplo, borrar todo lo mayor a un mega en /tmp.

Código:
find /tmp -size +3000k -exec rm -f {} \;

Por cierto si usas la versión GNU de **find** (y creo que todos los que usamos Linux la tenemos, compruébalo con *find --version*)), lo anterior también funciona directamente con la opción *-delete* :

Código:
find /tmp -size +3000k -delete (lo mismo que usar -exec con rm)