

Introducción a Linux

Módulo 2



Herramientas de búsqueda

Herramientas de búsqueda

FHS sirve como una **guía para ubicar y encontrar archivos y directorios**, pero eso no garantiza la posibilidad de encontrar cualquier archivo.



Linux nos proporciona varias herramientas de búsqueda. Las principales son:

- **which** muestra la ruta de los comandos usando la variable entorno PATH.
- **find** busca archivos en un árbol de directorios.
- **whereis** muestra la ruta de los comandos junto con las de sus páginas del manual.
- **locate** ofrece busca archivos indexados en una base datos en lugar del propio sistema de archivos.

Comando locate

Sintaxis

```
locate patrón
```

La búsqueda se realiza en un índice creado con el comando **updatedb**, busca los archivos cuyos nombres coincidan con uno o más patrones.

Buscar el patrón sh:

```
# locate zyxel  
/usr/share/wireshark/radius/dictionary.zyxel
```

El comando **locate** debe tener una base de datos reciente para la búsqueda. Esa base de datos debe ser actualizada periódicamente para incorporar los cambios en el sistema de archivos. Si la base de datos está muy desactualizada, el comando puede mostrar una advertencia como se ve a continuación:

```
# locate tcsh
```

```
locate: warning: database '/var/lib/slocate/slocate.db' is more than 8 days old
```

Para actualizarla hay que ejecutar el comando **updatedb**.



Comando updatedb

Sintaxis

```
updatedb [opciones]
```

Actualizar (o crear) la base de datos slocate en /var/lib/slocate/slocate.db.

Ejemplo para actualizar solo lo contenido en un grupo de directorios:

```
# updatedb -e "/dev /proc /home"
```

Hay parámetros especiales, con los que podemos hacer que realice la búsqueda por red o que no busque en determinados sistemas de archivos.

```
# cat /etc/updatedb.conf
PRUNEFS = "auto afs gfs gfs2 iso9660 sfs udf"
PRUNEPATHS = "/afs /media /net /sfs /tmp /udev
/var/spool/cups /var/spool/squid /var/tmp"
```

Comando whereis

Sintaxis

```
whereis [options] nombre de archivo
```

whereis localiza source/binarios y manuales para los archivos especificados.

Un ejemplo:

```
# whereis ls  
ls: /bin/ls /usr/share/man/man1p/ls.1p.gz /usr/share/man/man1/ls.1.gz
```

Comando which

Sintaxis

```
which comando
```

Determina la ubicación del comando y muestra la ruta completa del programa ejecutable que la *shell* lanzaría para ejecutarlo. Este sólo busca las rutas definidas en la variable PATH.

```
# which bash  
/bin/bash
```



Comando find

El comando **find** es una herramienta muy potente que permite buscar, entre otras cosas:

- En la jerarquía del sistema de archivos, dentro de un determinado directorio.
- Archivos por nombre.
- Archivos por tipo.
- Archivos por marcas de tiempo.
- Archivos por permisos.
- Archivos de acuerdo a su dueño.

Sintaxis

```
find [options] [path...] [expression]
```

-mount no va a seguir buscando en forma recursiva por directorios que fueron montados.


-maxdepth X se le indica cuál es la cantidad máxima de subdirectorios que se desea que recorra.

per-m para buscar por permisos (-perm 777)

Buscar por nombre

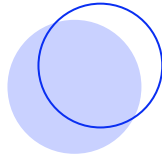
```
# find /home/rino/scripts/ -name "*.sh"  
/home/rino/scripts/vpn.sh  
/home/rino/scripts/irssi_notify.sh  
/home/rino/scripts/rino.sh
```

Es conveniente encerrar entre comillas para que la *shell* no reemplace automáticamente los metacaracteres en el directorio actual.



Buscar aquellos archivos que se hayan creado hace más de 7 días, y máximo un subdirectorio de profundidad

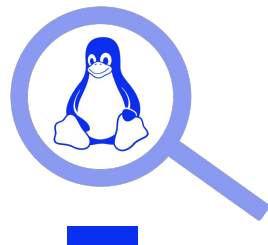
```
# find /tmp -maxdepth 1 -type f -daystart -ctime +7  
/tmp/soDumpciZ4eX  
/tmp/sakis3g.dialog.6712  
/tmp/salida.out  
/tmp/sakis3gz.6677.sakis3g
```



Buscar archivos .tmp sin importar mayúsculas o minúsculas y ejecutar una acción sobre los archivos encontrados

```
# find /home -iname "*.tmp" -exec rm -i '{}' \;
```

La cadena de texto '{}' es reemplazada por el archivo procesado por **find**, y el \; significa que no hay más argumentos para el comando después del **exec** (en este caso **rm**). Tanto las comillas simples como la contrabarra se ponen para que bash interprete de manera literal esos caracteres.



Buscar por tamaño

```
# find /usr -size +110M
/usr/lib/jvm/java-10-openjdk-10.0.2.13-1.fc28.x86_64/lib/modules
/usr/lib/locale/locale-archive
/usr/share/windows95/resources/app.asar.unpacked/src/images/windows95.img
/usr/local/share/.config/yarn/global/node_modules/puppeteer/.local-chromium/linux-579032/chrome-linux/chrome
```



Buscará los archivos con un tamaño mayor a 110 MB.

Más ejemplos

| | | |
|--------------------------------------|--|--|
| Buscar por nombre de archivo. | <code>find /etc -name "*.sh"</code> | Usar comillas al usar metacaracteres para evitar que la shell realice autocompletado de rutas. |
| Buscar por tipo y nombre de archivo. | <code>find / -type d -name images</code> | |
| Buscar por fecha de modificación. | <code>find /var -mtime -2</code> | Buscará todos los archivos que fueron modificados hace menos de 2 x 24 hs exactamente. |
| | <code>find /var -mtime +2</code> | Buscará todos los archivos que fueron modificados hace más de 2 x 24 hs exactamente. |
| | <code>find /var -mtime 2</code> | Buscará todos los archivos que fueron modificados hace 2 x 24hs exactamente. |

**¡Sigamos
trabajando!**