

# ACCESO A LOS SISTEMAS DE ARCHIVOS DE LINUX

RH124

Capítulo 14



Descripción general	
<b>Meta</b>	Acceder a sistemas de archivos existentes y examinarlos en un sistema con Red Hat Enterprise Linux.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Identificar la jerarquía del sistema de archivos.</li> <li>• Acceder al contenido de los sistemas de archivos.</li> <li>• Usar enlaces duros y enlaces simbólicos para crear múltiples nombres.</li> <li>• Buscar archivos en sistemas de archivos montados.</li> </ul>
<b>Secciones</b>	<ul style="list-style-type: none"> <li>• Identificación de dispositivos y sistemas de archivos (y práctica)</li> <li>• Montaje y desmontaje de sistemas de archivos (y práctica)</li> <li>• Creación de enlaces entre archivos (y práctica)</li> <li>• Búsqueda de archivos en el sistema (y práctica)</li> </ul>
<b>Trabajo de laboratorio</b>	<ul style="list-style-type: none"> <li>• Acceso a los sistemas de archivos de Linux</li> </ul>

# Identificación de dispositivos y sistemas de archivos

RH124

An abstract geometric design on a red background. It features several white lines of varying lengths and thicknesses. Some lines intersect at small white dots. There are also some larger, faint white circular shapes in the background. The overall composition is modern and minimalist.

# Objetivos

Tras finalizar esta sección, los estudiantes deberían poder identificar un directorio en la jerarquía de sistemas de archivos y el dispositivo de almacenamiento en el que está almacenado.

## Conceptos de la administración del almacenamiento

Un sistema de archivos es una estructura organizada de directorios y archivos que contienen datos que residen en un dispositivo de almacenamiento, como una partición o un disco físico. La jerarquía de sistemas de archivos abordada anteriormente reúne todos los sistemas de archivos en un árbol de directorios con una sola raíz: el directorio /. La ventaja de esta disposición es que la jerarquía existente puede extenderse en cualquier momento mediante la adición de una partición o un disco nuevo que contenga un sistema de archivos compatible a fin de añadir espacio en disco en cualquier parte del árbol del sistema de archivos. El proceso mediante el cual se añade un sistema de archivos nuevo al árbol de directorios existente se denomina *montaje*. El directorio en el que se monta el sistema de archivos nuevo se conoce como *punto de montaje*. Este concepto es muy diferente del empleado en un sistema Microsoft Windows, en el que un sistema de archivos nuevo se representa como una letra de unidad separada.

Los discos duros y los dispositivos de almacenamiento normalmente se dividen en fragmentos más pequeños llamados *particiones*. Una partición es una forma de compartimentar un disco. Sus diferentes partes pueden formatearse con diferentes sistemas de archivos o utilizarse con fines distintos. Por ejemplo, una partición puede contener un directorio de inicio de un usuario mientras que otra puede contener registros y datos del sistema. Si un usuario llena la partición del directorio de inicio con datos, la partición del sistema puede seguir teniendo espacio disponible. La colocación de datos en dos sistemas de archivos separados en dos particiones diferentes colabora con la planificación del almacenamiento de datos.

Los discos duros y los dispositivos de almacenamiento normalmente se dividen en fragmentos más pequeños llamados *particiones*. Una partición es una forma de compartimentar un disco. Sus diferentes partes pueden formatearse con diferentes sistemas de archivos o utilizarse con fines distintos. Por ejemplo, una partición puede contener un directorio de inicio de un usuario mientras que otra puede contener registros y datos del sistema. Si un usuario llena la partición del directorio de inicio con datos, la partición del sistema puede seguir teniendo espacio disponible. La colocación de datos en dos sistemas de archivos separados en dos particiones diferentes colabora con la planificación del almacenamiento de datos.

Los dispositivos de almacenamiento se representan con un tipo de archivo especial denominado *dispositivo de bloque*. El dispositivo de bloque se almacena en el directorio `/dev`. En Red Hat Enterprise Linux, el primer disco duro SCSI, PATA/SATA o USB detectado es `/dev/sda`, el segundo es `/dev/sdb`, y así sucesivamente. Este nombre representa el disco en su totalidad. La primera partición primaria en `/dev/sda` es `/dev/sda1`, la segunda es `/dev/sda2` y así sucesivamente.

Un listado extenso del archivo de dispositivo `/dev/vda` en `serverX` revela que su tipo de archivo especial es `b`, que significa "dispositivo de bloque":

```
[student@serverX ~]$ ls -l /dev/vda  
brw-rw----. 1 root disk 253, 0 Mar 13 08:00 /dev/vda
```



## nota

Una excepción son los discos duros en máquinas virtuales que generalmente se muestran como `/dev/vd<letter>` o `/dev/xvd<letter>`.

Otra manera de organizar discos y particiones es mediante la *administración de volúmenes lógicos* (LVM). Con LVM, uno o más dispositivos de bloque pueden agregarse a un grupo de almacenamiento denominado *grupo de volúmenes*. El espacio en disco se pone a disposición con uno o más *volúmenes lógicos*. Un volumen lógico es el equivalente a una partición que reside en un disco físico. Tanto el grupo de volúmenes como el volumen lógico tienen nombres que se asignan tras su creación. Para el grupo de volúmenes, existe un directorio con el mismo nombre que el grupo de volúmenes en el directorio `/dev`. Debajo de ese directorio, se ha creado un enlace simbólico con el mismo nombre que el volumen lógico. Por ejemplo, el archivo de dispositivo que representa el volumen lógico `mylv` en el grupo de volúmenes `myvg` es `/dev/myvg/mylv`.

# Examinar sistemas de archivos

Para obtener una descripción general sobre los puntos de montaje de sistemas de archivos y la cantidad de espacio libre disponible, ejecute el comando **df**. Cuando el comando **df** se ejecuta sin argumentos, arrojará un informe con el espacio en disco total, el espacio en disco usado y el espacio en disco libre en todos los sistemas de archivos regulares montados. Arrojará un informe sobre los sistemas locales y remotos, y el porcentaje de espacio en disco total que se está empleando.

Visualizar los sistemas de archivos y los puntos de montaje en la máquina serverX.

```
[student@serverX ~]$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/vda1        6240256 4003760   2236496   65% /
devtmpfs         950536      0    950536    0% /dev
tmpfs            959268      0    959188    1% /dev/shm
tmpfs            959268    2156    957112    1% /run
tmpfs            959268      0    959268    0% /sys/fs/cgroup
```

La partición en la máquina serverX muestra un sistema de archivos real, que está montado en /. Esto es frecuente en el caso de máquinas virtuales. Los dispositivos *tmpfs* y *devtmpfs* son sistemas de archivos en la memoria del sistema. Todos los archivos escritos en *tmpfs* o en *devtmpfs* desaparecen después de que se reinicia el sistema.

A fin de mejorar la legibilidad de los tamaños de los resultados, hay dos opciones diferentes *legibles por el ojo humano* : **-h** o **-H**. La diferencia entre estas dos opciones es que **-h** informará en KiB ( $2^{10}$ ), MiB ( $2^{20}$ ), o GiB ( $2^{30}$ ), mientras que la opción **-H** informará en unidades SI: KB ( $10^3$ ), MB ( $10^6$ ), GB ( $10^9$ ), etc. Los fabricantes de discos duros normalmente usan las unidades SI cuando anuncian sus productos.

Muestre un informe sobre los sistemas de archivos en la máquina serverX con todas las unidades convertidas a formato legible por el ojo humano:

```
[student@serverX ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1        6.0G  3.9G  2.2G  65% /
devtmpfs         929M      0  929M   0% /dev
tmpfs            937M   80K  937M   1% /dev/shm
tmpfs            937M  2.2M  935M   1% /run
tmpfs            937M      0  937M   0% /sys/fs/cgroup
```

Para obtener información más detallada sobre el espacio utilizado por un árbol de directorios en particular, puede usar el comando **du**. El comando **du** ofrece las opciones **-h** y **-H** para convertir el resultado a formato legible por el ojo humano. El comando **du** muestra el tamaño de todos los archivos en el árbol de directorios actual de modo recursivo.



Muestre un informe sobre el uso del disco para el directorio **/root** en serverX:

```
[root@serverX ~]# du /root
4 /root/.ssh
4 /root/.cache/dconf
4 /root/.cache
4 /root/.dbus/session-bus
4 /root/.dbus
0 /root/.config/ibus/bus
0 /root/.config/ibus
0 /root/.config
14024 /root
```

Muestre un informe sobre el uso del disco en formato legible por el ojo humano, para el directorio **/var/log** en serverX:

```
[root@serverX ~]# du -h /var/log
...
4.9M /var/log/sa
68K /var/log/prelink
0 /var/log/qemu-ga
14M /var/log
```



## Referencias

Páginas del manual: **df**(1), **du**(1)

# Práctica: Identificación de los dispositivos y sistemas de archivos

# Montaje y desmontaje de sistemas de archivos

RH124

# Objetivos

Tras finalizar esta sección, los estudiantes deberían poder acceder al contenido de sistemas de archivos mediante la adición y la eliminación de sistemas de archivos de la jerarquía de sistemas de archivos.

## Montaje manual de sistemas de archivos

Un sistema de archivos que reside en un dispositivo SATA/PATA o SCSI debe montarse manualmente para acceder a él. El comando **mount** permite que el usuario root monte manualmente un sistema de archivos. El primer argumento del comando **mount** especifica el sistema de archivos que se debe montar. El segundo argumento especifica el directorio de destino en el que se pone a disposición el sistema de archivos después de su montaje. El directorio de destino se conoce como punto de montaje.

El comando **mount** espera el argumento correspondiente al sistema de archivos en una de dos maneras diferentes:

- El archivo de dispositivo de la partición que contiene el sistema de archivos, que reside en **/dev**.
- El *UUID*, identificador único universal del sistema de archivos.



## **nota**

En la medida en que un sistema de archivos no se recree, el UUID no cambia. El archivo de dispositivo puede cambiar, por ejemplo, si se modifica el orden de los dispositivos o si se añaden dispositivos adicionales al sistema.

El comando **blkid** ofrece una descripción general de las particiones existentes con un sistema de archivos en ellas y el UUID del sistema de archivos, así como también el sistema de archivos utilizado para formatear la partición.

```
[root@serverX ~]# blkid  
/dev/vda1: UUID="46f543fd-78c9-4526-a857-244811be2d88" TYPE="xfs"
```



## nota

Un sistema de archivos puede montarse en un directorio existente. El directorio **/mnt** existe de manera predeterminada y proporciona un punto de entrada para los puntos de montaje. Se utiliza para el montaje manual de discos. Se recomienda crear un subdirectorio en **/mnt** y usarlo como punto de montaje, a menos que haya un motivo para montar el sistema de archivos en otra ubicación específica en la jerarquía del sistema de archivos.

Monte por archivo de dispositivo de la partición que contiene el sistema de archivos.

```
[root@serverX ~]# mount /dev/vdb1 /mnt/mydata
```

Monte el sistema de archivos por ID única universal, o UUID, del sistema de archivos.

```
[root@serverX ~]# mount UUID="46f543fd-78c9-4526-a857-244811be2d88" /mnt/mydata
```



## nota

Si el directorio que funciona como punto de montaje no está vacío, no se podrá acceder a los archivos que existen en ese directorio en la medida en que el sistema de archivos esté montado en él. Todos los archivos escritos en el directorio de punto de montaje terminan en el sistema de archivos montado en él.



# Desmontaje de sistemas de archivos

Para desmontar un sistema de archivos, el comando **umount** espera el punto de montaje como argumento.

Cambie al directorio **/mnt/mydata**. Intente desmontar el dispositivo montado en el punto de montaje **/mnt/mydata**. Ocurrirá un error.

```
[root@serverX ~]# cd /mnt/mydata
[root@serverX mydata]# umount /mnt/mydata
umount: /mnt/mydata: target is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
```

No se puede realizar el desmontaje si un proceso accede al punto de montaje. Para que el comando **umount** se ejecute correctamente, el proceso debe dejar de acceder al punto de montaje.

El comando **lsof** enumera todos los archivos abiertos y el proceso que accede a ellos en el directorio proporcionado. Resulta útil identificar los procesos que actualmente impiden un correcto desmontaje del sistema de archivos.

```
[root@serverX mydata]# lsof /mnt/mydata
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
bash	1593	root	cwd	DIR	253,2	6	128	/mnt/mydata
lsof	2532	root	cwd	DIR	253,2	19	128	/mnt/mydata
lsof	2533	root	cwd	DIR	253,2	19	128	/mnt/mydata

Una vez que se identifican los procesos, puede tomarse una medida, como esperar a que finalice el proceso o enviar una señal SIGTERM o SIGKILL al proceso. En este caso, basta con cambiar el directorio en funcionamiento actual por un directorio fuera del punto de montaje.

```
[root@serverX mydata]# cd  
[root@serverX ~]# umount /mnt/mydata
```

## nota

Una causa frecuente por la cual el sistema de archivos en el punto de montaje está ocupado es que el directorio en funcionamiento actual de un aviso de la shell se encuentra debajo del punto de montaje activo. El proceso que accede al punto de montaje es **bash**. El cambio a un directorio fuera del punto de montaje permite el desmontaje del dispositivo.

# Acceso a dispositivos de almacenamiento extraíbles

El entorno de escritorio gráfico monta automáticamente los medios extraíbles, como memorias y dispositivos flash USB, cuando se conectan. El punto de montaje para el medio extraíble es `/run/media/<user>/<label>`. El valor de `<user>` es el usuario registrado en el entorno gráfico. El valor de `<label>` es el nombre que se le asignó al sistema de archivos cuando se creó.



## Advertencia

Para extraer medios USB del sistema de manera segura, primero hay que desmontarlos y, luego, extraerlos físicamente de la ranura USB para sincronizar el sistema de archivos. La extracción de un dispositivo de almacenamiento USB sin desmontar el sistema de archivos en él puede ocasionar la pérdida de datos.



## Referencias

Páginas del manual: `mount(8)`, `umount(8)`, `lsfs(8)`

# Práctica: Montar y desmontar sistemas de archivos

# Creación de enlaces entre archivos

RH124



# Objetivos

Tras finalizar esta sección, los estudiantes deberían poder usar enlaces duros y blandos para que múltiples nombres apunten al mismo archivo.

## Administración de enlaces entre archivos

### Creación de enlaces duros

Un enlace duro es una nueva entrada en el directorio que hace referencia a un archivo existente en el sistema de archivos. Todos los archivos de un sistema de archivos tienen un enlace duro nuevo de manera predeterminada. En lugar de copiar un archivo, puede crearse un enlace duro que haga referencia al mismo archivo y así ahorrar espacio. Un enlace duro nuevo debe tener un nombre de archivo diferente si se crea en el mismo directorio que el enlace duro existente o debe residir en un directorio distinto. Todos los enlaces duros que apuntan al mismo archivo tienen iguales permisos, valor de enlace, propiedades de usuario o grupo, sellos de fecha y hora, y contenido de archivo. Los enlaces duros que apuntan al contenido del mismo archivo deben estar en el mismo sistema de archivos.

El comando **ls -l** muestra el valor del enlace posterior a los permisos y anterior al propietario de un archivo.

```
[root@serverX ~]# echo "Hello World" > newfile.txt
[root@serverX ~]# ls -l newfile.txt
-rw-r--r--. 1 root root 0 Mar 11 19:19 newfile.txt
```

El comando **ln** crea enlaces duros nuevos a archivos existentes. El comando espera un archivo existente como el primer argumento, seguido por uno o más enlaces duros adicionales. Los enlaces duros pueden residir en cualquier parte siempre que estén en el mismo sistema de archivos que el archivo existente. Después de que se crea un enlace duro nuevo, no existe manera de saber cuál de los enlaces duros existentes es el original.

Cree un enlace duro **newfile-link2.txt** para el archivo existente **newfile.txt** en el directorio **/tmp**.

```
[root@serverX ~]# ln newfile.txt /tmp/newfile-hlink2.txt
[root@serverX ~]# ls -l newfile.txt /tmp/newfile-hlink2.txt
-rw-rw-r--. 2 root root 12 Mar 11 19:19 newfile.txt
-rw-rw-r--. 2 root root 12 Mar 11 19:19 newfile-hlink2.txt
```

Incluso si se elimina el archivo original, el contenido del archivo continúa estando disponible siempre y cuando exista un enlace duro como mínimo.

```
[root@serverX ~]# rm -f newfile.txt
[root@serverX ~]# ls -l /tmp/newfile-link2.txt
-rw-rw-r--. 1 root root 12 Mar 11 19:19 /tmp/newfile-link2.txt
[root@serverX ~]# cat /tmp/newfile-link2.txt
Hello World
```

## Importante

Todos los enlaces duros que hacen referencia al mismo archivo tienen iguales permisos, conteo de enlace, propiedades de usuario o grupo, sellos de fecha y hora, y contenido de archivo. Si se modifica algún dato en un enlace duro, todos los demás enlaces duros que apuntan al mismo archivo también mostrarán el dato nuevo.



## Creación de enlaces blandos

El comando `ln -s` permite crear un enlace blando, que también se conoce como "enlace simbólico". Un enlace blando no es un archivo regular, sino un tipo de archivo especial que apunta a un archivo o a un directorio existente. A diferencia de los enlaces duros, los enlaces blandos pueden apuntar a un directorio, y el objetivo al que apunta un enlace blando puede estar en un sistema de archivos diferente.

```
[root@serverX ~]# ln -s /root/newfile-link2.txt /tmp/newfile-symlink.txt
[root@serverX ~]# ls -l newfile-link2.txt /tmp/newfile-symlink.txt
lrwxrwxrwx. 1 root root 11 Mar 11 20:59 /tmp/newfile-symlink.txt -> /root/newfile-link2.txt
-rw-rw-r--. 1 root root 12 Mar 11 19:19 newfile-link2.txt
```

Cuando se elimina el archivo original, el enlace blando sigue apuntando al archivo, pero el destino desaparece. Un enlace blando que apunta a un archivo que falta recibe el nombre de "enlace blando colgante".

```
[root@serverX ~]# rm -f newfile-link2.txt
[root@serverX ~]# ls -l /tmp/newfile-symlink.txt
lrwxrwxrwx. 1 root root 11 Mar 11 20:59 /tmp/newfile-symlink.txt -> newfile-link2.txt
[root@serverX ~]# cat /tmp/newfile-symlink.txt
cat: /tmp/newfile-symlink.txt: No such file or directory
```

Un enlace blando puede apuntar a un directorio. El enlace blando funciona como un directorio. Si cambia el directorio del enlace blando con el comando `cd`, obtendrá el funcionamiento esperado.

Cree un enlace blando `/root/configfiles` que apunte al directorio `/etc`.

```
[root@serverX ~]# ln -s /etc /root/configfiles  
[root@serverX ~]# cd /root/configfiles  
[root@serverX configfiles]# pwd  
/root/configfiles
```



## Referencias

Página del manual (1)[ln](#)

# Práctica: Creación de enlaces entre archivos

# Localización de archivos en el sistema

RH124

An abstract geometric design on a red background. It features several thin white lines that intersect at small white dots. One line runs diagonally from the bottom left towards the top right. Another line runs diagonally from the bottom left towards the top right, intersecting the first line. A third line runs diagonally from the bottom right towards the top left, intersecting the other two. There are also some faint, larger circular shapes in the background.

# Objetivos

Tras finalizar esta sección, los estudiantes deberían poder buscar archivos en los sistemas de archivos montados usando los comandos **find** y **locate**.

## Herramientas para buscar archivos

Un administrador de sistemas necesita herramientas para buscar archivos que coincidan con ciertos criterios en el sistema de archivos. En esta sección, se analizan dos comandos que pueden buscar archivos en el sistema de archivos. El comando **locate** busca una base de datos generada previamente para nombres de archivos o rutas de archivos y arroja los resultados instantáneamente. El comando **find** busca el sistema de archivos en tiempo real mediante un rastreo del sistema de archivos.

# Localización de archivos por nombre con **locate**

El comando **locate** arroja resultados de búsqueda en función de la ruta o el nombre de archivos de la base de datos de **locate**. La base de datos almacena información de la ruta y el nombre de archivos.

Al buscar entradas como un usuario regular, los resultados solo se arrojan en los casos en que el usuario que solicita la búsqueda mediante **locate** tenga permisos de lectura en el árbol del directorio que contiene el elemento correspondiente.

Busque archivos con "passwd" en el nombre o la ruta en árboles de directorio legibles por el usuario student en serverX.

```
[student@serverX ~]$ locate passwd
/etc/passwd
/etc/passwd-
/etc/pam.d/passwd
/etc/security/opasswd
/usr/bin/gpasswd
/usr/bin/grub2-mkpasswd-pbkdf2
/usr/bin/lppasswd
/usr/bin/passwd
/usr/bin/userpasswd
/usr/bin/vino-passwd
/usr/bin/vncpasswd
```

Los resultados se presentan incluso cuando la ruta o el nombre de archivo es solo una coincidencia parcial con la consulta de búsqueda.

```
[root@serverX ~]# locate image
/home/myuser/boot.image
/home/someuser/my_family_image.png
/home/student/myimages-vacation/picture.png
```

La opción **-i** realiza una búsqueda que distingue entre mayúsculas y minúsculas. Con esta opción, todas las combinaciones posibles de letras en mayúsculas y minúsculas coinciden con la búsqueda.

```
[student@serverX ~]$ locate -i messages
...
/usr/share/vim/vim74/lang/zh_TW/LC_MESSAGES
/usr/share/vim/vim74/lang/zh_TW/LC_MESSAGES/vim.mo
/usr/share/vim/vim74/lang/zh_TW.UTF-8/LC_MESSAGES
/usr/share/vim/vim74/lang/zh_TW.UTF-8/LC_MESSAGES/vim.mo
/usr/share/vim/vim74/syntax/messages.vim
/usr/share/vim/vim74/syntax/msmessages.vim
/var/log/messages
```

La opción **-n** limita el número de resultados de búsqueda arrojados por **locate**. El siguiente ejemplo limita los resultados de búsqueda arrojados por **locate** a las primeras cinco coincidencias.

```
[student@serverX ~]$ locate -n 5 snow.png
/usr/share/icons/HighContrast/16x16/status/weather-snow.png
/usr/share/icons/HighContrast/22x22/status/weather-snow.png
/usr/share/icons/HighContrast/24x24/status/weather-snow.png
/usr/share/icons/HighContrast/256x256/status/weather-snow.png
/usr/share/icons/HighContrast/32x32/status/weather-snow.png
```



## nota

La base de datos **locate** se actualiza automáticamente todos los días. El usuario **root** puede realizar una actualización de la base de datos con el comando **updatedb**.

```
[root@serverX ~]# updatedb
```



# Búsqueda de archivos con find

El comando **find** realiza una búsqueda en tiempo real en los sistemas de archivos locales para encontrar archivos que coincidan con los criterios de los argumentos de la línea de comandos. El comando **find** busca archivos del sistema de archivos como su cuenta de usuario. El usuario que invoca el comando **find** debe tener permiso de lectura y ejecución en un directorio para examinar su contenido.

El primer argumento para el comando **find** es el directorio en que se realizará la búsqueda. Si el argumento del directorio se omite, **find** comenzará la búsqueda en el directorio actual y buscará coincidencias en todos los subdirectorios.

Para buscar el directorio de inicio del usuario student, dé a **find** un directorio de inicio **/home/student**. Para buscar en todo el sistema, proporcione un directorio de inicio de **/**.

## nota

**find** posee una gran cantidad de opciones para describir exactamente qué tipo de archivo se debe buscar. Las búsquedas se pueden basar en el nombre del archivo, el tamaño del archivo, el sello de tiempo de la última modificación y otras características de archivos en cualquier combinación.

La opción **-name** seguida del nombre de un archivo busca archivos que coincidan con el nombre de archivo dado y arroja coincidencias exactas. Para buscar archivos denominados **sshd\_config** en el directorio **/** y todos los subdirectorios en **serverX**, ejecute lo siguiente:

```
[root@serverX ~]# find / -name sshd_config  
/etc/ssh/sshd_config
```

Los comodines están disponibles para buscar el nombre de un archivo y arroja todos los resultados que son coincidencias parciales. Al usar comodines, es importante poner entre comillas el nombre del archivo para evitar que el terminal interprete el comodín.

En el siguiente ejemplo, se buscan archivos en el directorio **/** en **serverX** que finalicen en **.txt**:

```
[root@serverX ~]# find / -name '*.txt'  
/etc/pki/nssdb/pkcs11.txt  
/etc/brltty/brl-lt-all.txt  
/etc/brltty/brl-mb-all.txt  
/etc/brltty/brl-md-all.txt  
/etc/brltty/brl-mn-all.txt  
...
```

Para buscar archivos en `/etc/` que contengan **pass** en cualquier parte del nombre en `serverX`, ejecute lo siguiente:

```
[root@serverX ~]# find /etc -name '*pass*'
/etc/passwd
/etc/passwd-
/etc/fonts/conf.d/60-overpass.conf
/etc/selinux/targeted/modules/active/modules/passenger.pp
/etc/security/opasswd
/etc/pam.d/passwd
/etc/pam.d/password-auth-ac
/etc/pam.d/password-auth
/etc/pam.d/gdm-password
```

Para realizar una búsqueda que no distinga entre mayúsculas y minúsculas de un nombre de archivo determinado, use la opción **-iname**, seguida del nombre del archivo que desea buscar. Para realizar una búsqueda que no distinga entre mayúsculas y minúsculas de archivos que tengan **messages** en el nombre, en el directorio `/` en `serverX`, ejecute lo siguiente:

```
[root@serverX ~]# find / -iname '*messages*'
/var/log/messages
/usr/lib64/python2.7/site-packages/orca/notification_messages.py
/usr/lib64/python2.7/site-packages/orca/notification_messages.pyc
/usr/lib64/python2.7/site-packages/orca/notification_messages.pyo
/usr/share/locale/aa/LC_MESSAGES
/usr/share/locale/ab/LC_MESSAGES
/usr/share/locale/ace/LC_MESSAGES
```

**find** puede buscar archivos en base a la propiedad o los permisos. Las opciones útiles al buscar por propietario son **-user** y **-group**, que buscan por nombre, y **-uid** y **-gid**, que buscan por ID.

Busque archivos de propiedad del usuario *student* en el directorio **/home/student** en serverX.

```
[student@serverX ~]$ find -user student
.
./.bash_logout
./.bash_profile
./.bashrc
./.ssh
...
```

Busque archivos de propiedad del grupo *estudiante* en el directorio **/home/student** en serverX.

```
[student@serverX ~]$ find -group student
.
./.bash_logout
./.bash_profile
./.bashrc
./.ssh
...
```

Busque archivos de propiedad del usuario con la ID *1000* en el directorio **/home/student** en serverX.

```
[student@serverX ~]$ find -uid 1000
.
./.bash_logout
./.bash_profile
./.bashrc
./.ssh
...
```

Busque archivos de propiedad del grupo con ID *1000* en el directorio */home/student* en *serverX*.

```
[student@serverX ~]$ find -gid 1000
.
./bash_logout
./bash_profile
./bashrc
./ssh
...
```

Busque archivos de propiedad del usuario *root* y el grupo *mail* en la máquina *serverX*.

```
[root@serverX ~]# find / -user root -group mail
/var/spool/mail
/var/spool/mail/root
```

Se utiliza la opción **-perm** para buscar archivos con una serie de permisos particulares. Los permisos se pueden describir como valores octales, con alguna combinación de 4, 2 y 1 para lectura, escritura y ejecución. Los permisos deben estar precedidos por el signo */* o el signo *-*.

Un permiso numérico precedido por */* coincidirá con archivos que tengan al menos un bit de usuario, grupo u otro, para esa serie de permisos. Un archivo con permisos **r--r--r--**

no coincide con /222, pero uno con `rw-r--r--` sí. Un signo menos - antes de un permiso significa que las tres instancias de ese bit deben estar activadas; por lo tanto, ningún ejemplo anterior coincidirá, pero algo como `rw-rw-rw-` sí lo hará.

Para utilizar un ejemplo más complejo, el siguiente comando debe coincidir con cualquier archivo para el cual el usuario tiene permisos de lectura, escritura y ejecución, los miembros del grupo tienen permisos de lectura y escritura, y los demás tienen acceso de solo lectura:

```
[root@serverX ~]# find /home -perm 764
```

Para que coincidan los archivos para los cuales el usuario tiene al menos permisos de escritura y ejecución, y el grupo tiene por lo menos permisos de lectura y los demás tienen por lo menos acceso de lectura:

```
[root@serverX ~]# find /home -perm -324
```

Para que coincidan los archivos para los cuales el usuario tiene permisos de lectura, o el grupo tiene por lo menos permisos de lectura o los demás tienen por lo menos acceso de lectura:

```
[root@serverX ~]# find /home -perm /442
```

Cuando se utiliza con `/` o bien `-`, un valor de `0` funciona como un comodín, ya que significa "un permiso de por lo menos nada".

Para que coincida con cualquier archivo en el directorio `/home/student` para el cual los demás tienen al menos acceso de lectura en `serverX`, ejecute lo siguiente:

```
[student@serverX ~]$ find -perm -004
```

Encuentre todos los archivos en el directorio `/home/student` donde *other* tiene permisos de escritura en `serverX`.

```
[student@serverX ~]$ find -perm -002
```

El comando `find` puede buscar archivos que coincidan con un tamaño especificado con la opción `-size`, seguida de un valor numérico y la unidad.

Las unidades que se usarán con la opción `-size` son las siguientes:

- k, para kilobyte
- M, para megabyte
- G, para gigabyte

Busque archivos con un tamaño de *exactamente* 10 megabytes.

```
[student@serverX ~]$ find -size 10M
```

Busque archivos con un tamaño *mayor* que 10 gigabytes.

```
[student@serverX ~]$ find -size +10G
```

Detalle todos los archivos con un tamaño *menor* que 10 kilobytes.

```
[student@serverX ~]$ find -size -10k
```

## Importante

Los modificadores de la unidad **-size** redondean todo para arriba a unidades enteras. Por ejemplo, **find -size 1M** mostrará archivos de un tamaño menor que 1 MB porque redondea todos los archivos para arriba a 1 MB.



La opción **-mmin**, seguida de la hora en minutos, busca todos los archivos para los cuales se ha cambiado su contenido exactamente en el momento dado en el pasado.

Para encontrar todos los archivos para los cuales se había modificado su contenido exactamente hace 120 minutos en serverX, ejecute:

```
[root@serverX ~]# find / -mmin 120
```

El modificador **+** delante de la cantidad de minutos busca todos los archivos en **/** que han sido modificados hace más de 200 minutos.

```
[root@serverX ~]# find / -mmin +200
```

El modificador **-** cambia la búsqueda para buscar todos los archivos en el directorio **/** que han sido modificados hace menos de 150 minutos.

```
[root@serverX ~]# find / -mmin -150
```

La opción **-type** limita el alcance de la búsqueda a un tipo de archivo dado, como lo siguiente:

- **f**, para archivo regular
- **d**, para directorio
- **l**, para enlace simbólico
- **b**, para dispositivo de bloques

Busque todos los directorios en la carpeta **/etc** en serverX.

```
[root@serverX ~]# find /etc -type d
/etc
/etc/tmpfiles.d
/etc/systemd
/etc/systemd/system
/etc/systemd/system/getty.target.wants
...
```

```
[root@serverX ~]# find / -type l
```

Genere una lista de dispositivos de bloques en el directorio **/dev** en serverX:

```
[root@serverX ~]# find /dev -type b  
/dev/vda1  
/dev/vda
```

La opción **-links** seguida de un número busca todos los archivos que tienen un determinado conteo de enlaces duros. El número puede ser precedido por un modificador **+** para buscar archivos con un conteo más alto que el conteo de enlaces físicos dado. Si el número es precedido por un modificador **-**, la búsqueda se limita a todos los archivos con un conteo de enlaces físicos que sea menor que el número dado.

Busque todos los archivos regulares con más de un enlace físico en la máquina serverX.

```
[root@serverX ~]# find / -type f -links +1
```

## Referencias

Páginas del manual: **locate(1)**, **updatedb(1)**, **find(8)**

# Práctica: Búsqueda de archivos en el sistema