

Cómo configurar fstab y lilo.conf con nombres persistentes

¿Alguna vez ha tenido el orden de cambio de sus unidades /dev/sda y /dev/sdb? ¿Conectar un dispositivo USB antes de arrancar su computadora causó un pánico en el kernel? ¿Con frecuencia desenchufa o conecta dispositivos nuevos en su computadora, cambiando así el orden de los nombres de los dispositivos? Si ha respondido afirmativamente a alguno de ellos, podría ser un buen candidato para utilizar nombres persistentes en su gestor de arranque y fstab.

¿Qué es el nombramiento persistente?

El nombramiento persistente le permite hacer referencia a su unidad en forma fija. Hay varios métodos diferentes disponibles para la asignación de nombres persistentes y cubriremos lo siguiente: ID, etiqueta y UUID, así como PartLabel y PartUUID basados en GPT. Las más utilizadas son las etiquetas y los UUID, principalmente porque han existido durante mucho tiempo. Las tablas de partición GPT introdujeron una versión actualizada de etiquetas y UUID denominados partlabel y partuuid que se almacenan directamente en la tabla de partición. Sin embargo, a partir del kernel 3.8, PartUUID se ha expandido para generar identificadores únicos más cortos (UID) para las particiones MBR, permitiendo esencialmente que las unidades MBR usen PARTUUID.

Ahora, vamos a profundizar en los diversos métodos:

ID

/dev/disk/by-id/ es una carpeta generada dinámicamente bajo el sistema de archivos /dev/. Esto cambiará dependiendo de los dispositivos que estén conectados a la computadora. Las identificaciones generalmente están compuestas por el número de modelo del disco duro, el número de serie y las particiones específicas. `ls -la /dev/disk/by-id/` mostrará todos los ID y el dispositivo al que apunta bajo el sistema de archivos /dev/ (por ejemplo, /dev/sda, /dev/sdc3, etc).

```
slackware@darkstar:~$ ls -la /dev/disk/by-*
/dev/disk/by-id:
total 0
drwxr-xr-x 2 root root 680 Jul 31 13:17 ./
drwxr-xr-x 8 root root 160 Jul 31 13:17 ../
lrwxrwxrwx 1 root root 9 Jul 31 17:18 ata-Crucial_CT480M500SSD1_13440956798B
-> ../../sda
lrwxrwxrwx 1 root root 10 Jul 31 17:18 ata-
Crucial_CT480M500SSD1_13440956798B-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 Jul 31 17:18 ata-
Crucial_CT480M500SSD1_13440956798B-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 Jul 31 17:18 ata-
Crucial_CT480M500SSD1_13440956798B-part3 -> ../../sda3
lrwxrwxrwx 1 root root 9 Jul 31 17:18 ata-WDC_WD1001FALS-00J7B1_WD-
```

```
WMATV1953756 -> ../../sdb
lrwxrwxrwx 1 root root 10 Jul 31 17:18 ata-WDC_WD1001FALS-00J7B1_WD-
WMATV1953756-part1 -> ../../sdb1
lrwxrwxrwx 1 root root 9 Jul 31 17:18 ata-WDC_WD40EZRX-00SPEB0_WD-
WCC4E1021582 -> ../../sdc
lrwxrwxrwx 1 root root 10 Jul 31 17:18 ata-WDC_WD40EZRX-00SPEB0_WD-
WCC4E1021582-part1 -> ../../sdc1
```

Por lo tanto, si siempre desea hacer referencia a la unidad Crucial (solo la unidad, no las particiones dentro de ella), puede usar el nombre de archivo `/dev/disk/by-id/ ata-Crucial_CT480M500SSD1_13440956798B` . (Verá un ejemplo de esto en la sección `lilo.conf` a continuación). Esto siempre apuntará a esa unidad, independientemente de si es `/dev/sda` o `/dev/sdq` . Para hacer referencia a una partición específica, simplemente haga coincidir el nombre del dispositivo actual (`/dev/sdb1`) con su ID asociado, `/dev/disk/by-id/ ata-WDC_WD40EZRX-00SPEB0_WD-WCC4E1050616-part1` .

Label

Las etiquetas (label) son un nombre opcional configurado por el usuario para la unidad. La mayoría de los sistemas de archivos admiten esto, pero no siempre se crean automáticamente. Todas las etiquetas se pueden encontrar en `/dev/disk/by-label/` y se pueden ver fácilmente usando `ls -la /dev/disk/by-label/`.

```
/dev/disk/by-label:
total 0
drwxr-xr-x 2 root root 80 Jul 31 13:17 ./
drwxr-xr-x 8 root root 160 Jul 31 13:17 ../
lrwxrwxrwx 1 root root 10 Jul 31 17:18 Backup -> ../../sdc1
lrwxrwxrwx 1 root root 10 Jul 31 17:18 Data -> ../../sdb1
```

Las etiquetas pueden ser bastante convenientes sobre las ID debido a su capacidad para ser extremadamente cortas y memorables. Sin embargo, si no tiene cuidado o si mueve los discos entre sistemas, puede tener problemas con nombres duplicados que causan conflictos. Tanto `lilo` como `fstab` admiten el uso de `LABEL = "Backup"` para hacer referencia directamente a la unidad en lugar de tener que usar `/dev/disk/by-label/ Backup`, lo que hace que esta sea una opción mucho más sencilla para hacer referencia a las unidades.

Si desea agregar una etiqueta a su partición, [ArchWiki](#) esta página tiene instrucciones para la mayoría de los sistemas de archivos.

UUID

Los UUID (IDentificadores Únicos Universales) son un identificador único generado para una partición. Todos los sistemas de archivos de Linux generarán UUID cuando se formatean. Los sistemas de archivos FAT y NTFS no admiten UUID, sin embargo, se utiliza un UID corto en lugar de un UUID normal. Todos los UUID se pueden encontrar en `/dev/disk/by-uuid/` y se pueden ver usando `ls -la /dev/disk/by-uuid/`

```
/dev/disk/by-uuid:
total 0
drwxr-xr-x 2 root root 200 Jul 31 13:17 ./
drwxr-xr-x 8 root root 160 Jul 31 13:17 ../
lrwxrwxrwx 1 root root 10 Jul 31 17:18 226f1db3-2f14-4862-9998-8be1e8f7cdc9
-> ../../sda3
lrwxrwxrwx 1 root root 10 Jul 31 17:18 23bce2c2-996d-449e-89cc-0e5029cc6d8d
-> ../../sda2
lrwxrwxrwx 1 root root 10 Jul 31 17:18 bc4538e5-9387-4e5c-877b-d15744acb6ce
-> ../../sda1
lrwxrwxrwx 1 root root 10 Jul 31 17:18 cd554cf8-c346-4bd5-ba6a-84ac99af571f
-> ../../sdc1
lrwxrwxrwx 1 root root 10 Jul 31 17:18 2ef6e776-c2eb-4e56-bc7e-dfe083559d8f
-> ../../sdb1
lrwxrwxrwx 1 root root 10 Jul 31 17:18 126AC3506AC32EF3 -> ../../sdq1
#Example NTFS UID -- not really in my system
```

También se puede hacer referencia a los UUID utilizando solo UUID = 226f1db3-2f14-4862-9998-8be1e8f7cdc9 en sus archivos `/etc/lilo.conf` y `/etc/fstab`.

GPT (PartLabel and PartUUID)

El **MBR** (Master Boot Record) ha sido la tabla de partición primaria para los discos duros durante las últimas décadas (se introdujo en 1983). Sin embargo, no se puede utilizar en discos de más de 2 TB. Ahí es donde entra **GPT** (Tabla de particiones GUID). Esto se desarrolló como parte de **UEFI** espec. Pero se puede usar sin usar UEFI.

Además de las limitaciones del MBR, también existen limitaciones para el uso de UUID y etiquetas, principalmente porque si usa los de su cargador de arranque, debe usar un **initrd**. Si bien Pat recomienda usar un **initrd**, muchos usuarios prefieren usar un kernel que no requiere uno (ya sea usando el kernel enorme o construyendo un kernel personalizado con el soporte de hardware necesario compilado). La especificación de GPT agregó **PartLabel** y **PARTUUID** como una forma de guardar esa información directamente en la tabla de particiones (“part” significa partición). Si bien **PartLabel** y **PartUUID** están incluidos en la especificación GPT, el kernel solo admite el inicio de **PARTUUID**, por lo que no puede usar **PartLabel** en sus cargadores de arranque. Si desea usar **PartUUID**, no necesita ningún **initrd** para poder leer estos datos, lo que hace que sea mucho más fácil de usar en sistemas que aún no usan **initrd**. Inicialmente, dado que **PARTUUID** se introdujo como parte de la especificación de GPT, no se pudo hacer referencia a los dispositivos basados en MBR utilizando **PARTUUID**; sin embargo, el soporte básico para **PartUUID** en particiones MBR se agregó a partir del kernel 3.8 de Linux, por lo que ahora se puede hacer referencia a cualquier unidad con su “**PARTUUID**”.

PartLabel

Esto es casi idéntico a la opción de etiqueta normal. Se encuentra en `/dev/disk/by-partlabel/` y se puede ver con `ls -la /dev/disk/by-partlabel`. **PARTLABEL** solo se puede utilizar en el `fstab`. Debido a la falta de soporte en el kernel, necesitará usar una de las otras convenciones de nombres persistentes en su `lilo.conf`

```
/dev/disk/by-partlabel:
total 0
drwxr-xr-x 2 root root 60 Jul 31 17:18 ./
drwxr-xr-x 8 root root 160 Jul 31 13:17 ../
lrwxrwxrwx 1 root root 10 Jul 31 17:18 Linux\x20filesystem -> ../../sdc1
```

Además, esto solo mostrará PARTLABELs oficiales. No mostrará ninguna etiqueta de los discos que utilizan MBR. Si necesita verlos, puede usar `blkid`.

```
slackware@darkstar:~$ su -
Password:
root@darkstar:~# blkid
/dev/sdc1: LABEL="Backup" UUID="cd554cf8-c346-4bd5-ba6a-84ac99af571f"
TYPE="ext4" PARTLABEL="Linux filesystem" PARTUUID="62c372af-e868-4571-8ab4-
db612c0fb38f"
/dev/sdb1: LABEL="Data" UUID="2ef6e776-c2eb-4e56-bc7e-dfe083559d8f"
TYPE="ext4" PARTUUID="71b70c68-01"
/dev/sda1: UUID="bc4538e5-9387-4e5c-877b-d15744acb6ce" TYPE="swap"
PARTUUID="6f47c81b-01"
/dev/sda2: UUID="23bce2c2-996d-449e-89cc-0e5029cc6d8d" TYPE="ext4"
PARTUUID="6f47c81b-02"
/dev/sda3: UUID="226f1db3-2f14-4862-9998-8be1e8f7cdc9" TYPE="ext4"
PARTUUID="6f47c81b-03"
```

Como puede ver en el caso de `/dev/sdc1`, puede tener una LABEL y PARTLABEL diferente. Esto se debe a que estos se almacenan de forma independiente el uno del otro. LABEL es un rasgo del sistema de archivos, mientras que PARTLABEL es un rasgo de partición. También tenga en cuenta que, en función de cómo creé mi sistema de archivos, tiene un PARTLABEL genérico del "sistema de archivos Linux", por lo que si tuviera que crear un nuevo sistema de archivos en una nueva unidad sin cambiar nada, tendría dos unidades con la misma PARTLABEL. Esto no es un problema a menos que intente hacer referencia al disco como un PARTLABEL.

Si desea agregar o cambiar su PARTLABEL, puede hacerlo usando `gdisk` o `cgdisk`.

PartUUID

Esto es similar a PartLabel en que es casi idéntico a su contraparte "Part" (UUID regular). Este UUID se genera automáticamente cada vez que crea/redimensiona/mueve una partición. Este será un UUID diferente al proporcionado por el sistema de archivos. Todos los PartUUID se pueden encontrar en `/dev/disk/by-partuuid/` y se pueden ver usando `ls -la /dev/disk/by-partuuid/`.

```
/dev/disk/by-partuuid:
total 0
drwxr-xr-x 2 root root 120 Jul 31 13:17 ./
drwxr-xr-x 8 root root 160 Jul 31 13:17 ../
lrwxrwxrwx 1 root root 10 Jul 31 17:18 468a6e66-39e3-4072-8997-0550dd12ad8e
-> ../../sdb1
```

Como puede ver, esto es como PartLabel solo muestra los PartUUID oficiales en los discos GPT. Si necesita ver el PartUUID generado por el sistema para particiones MBR, necesita usar blkid.

```
slackware@darkstar:~$ su -
Password:
root@darkstar:~# blkid
/dev/sdc1: LABEL="Backup" UUID="cd554cf8-c346-4bd5-ba6a-84ac99af571f"
TYPE="ext4" PARTLABEL="Linux filesystem" PARTUUID="62c372af-e868-4571-8ab4-
db612c0fb38f"
/dev/sdb1: LABEL="Data" UUID="2ef6e776-c2eb-4e56-bc7e-dfe083559d8f"
TYPE="ext4" PARTUUID="71b70c68-01"
/dev/sda1: UUID="bc4538e5-9387-4e5c-877b-d15744acb6ce" TYPE="swap"
PARTUUID="6f47c81b-01"
/dev/sda2: UUID="23bce2c2-996d-449e-89cc-0e5029cc6d8d" TYPE="ext4"
PARTUUID="6f47c81b-02"
/dev/sda3: UUID="226f1db3-2f14-4862-9998-8be1e8f7cdc9" TYPE="ext4"
PARTUUID="6f47c81b-03"
```

Para sistemas de archivos que no son GPT, el PARTUUID se genera a partir del Identificador de disco, seguido del número de partición. Puedes ver tu identificador de disco usando fdisk.

```
root@darkstar:~# fdisk -l /dev/sda | grep identifier
Disk identifier: 0x6f47c81b
```



Debido a la novedad del soporte de GPT, lilo no tiene soporte directo para llamarlo, más bien necesita usar una opción de complemento (vea la sección lilo.conf a continuación para obtener más información). Para PartUUIDs en discos MBR, la única vez que cambiarán es si se crean particiones adicionales. Tenlo en cuenta si los estás usando en tu fstab y/o lilo.conf.

fstab

Su `/etc/fstab` es el archivo más fácil de cambiar. Simplemente reemplace cualquier entrada que apunte a los nombres de su dispositivo con `TYPE=value` (`TYPE` puede ser `LABEL`, `UUID`, `PARTLABEL`, `PARTUUID`). Los nombres de dispositivos originales se enumeran como comentarios para su referencia, no es necesario que estén en el archivo final. (Personalmente, ya que me gusta que se vea bien y ordenado, haré que todo esté espaciado uniformemente).

```
slackware@darkstar:~$ cat /etc/fstab
# /dev/sda1
UUID=bc4538e5-9387-4e5c-877b-d15744acb6ce          swap          swap
defaults,discard                                0 0
# /dev/sda2
UUID=23bce2c2-996d-449e-89cc-0e5029cc6d8d        /              ext4
defaults,noatime,discard                          0 1
# /dev/sda3
PARTUUID="6f47c81b-03"                            /home         ext4
```

```
defaults,noatime,discard          0 2
# /dev/sdb1
LABEL=Data                        /share/Data      ext4
defaults                          0 2
# /dev/sdc1
PARTLABEL="Linux filesystem"     /share/Backup    ext4
defaults                          0 2

#/dev/cdrom                       /mnt/cdrom      auto
noauto,owner,ro,comment=x-gvfs-show 0 0
devpts                             /dev/pts        devpts
gid=5,mode=620                    0 0
proc                               /proc           proc
defaults                          0 0
tmpfs                              /dev/shm        tmpfs
defaults                          0 0
```

lilo.conf

Su `/etc/lilo.conf` requiere un poco más de precisión para cambiar. No todo está correctamente soportado como las opciones de `fstab`, por lo que necesitamos incluir algunas soluciones alternativas, dependiendo de lo que quieras usar.

Para comenzar, necesitamos configurar la parte superior de su `/etc/lilo.conf`. Esto incluye principalmente la entrada de arranque. Esto es lo que usa `lilo` para determinar dónde escribir el código del bootloader en la tabla de particiones. Solo se hace referencia a esto cuando ejecuta `lilo` para instalar el bootloader, por lo que no se verá tan afectado como otras unidades si se mueven. Sin embargo, si ya se ha tomado la molestia de llegar tan lejos, también debería asegurarse de no tener ningún problema al hacer referencia a su unidad utilizando nombres permanentes.

Para hacer esto, deberíamos reemplazar `/dev/sda` (o cualquiera que sea su unidad de arranque predeterminada) con una referencia que no cambie. Ya que necesitamos hacer referencia a la unidad en sí, y no a particiones específicas, necesitamos usar el método de ID. Una vez que encuentre la ID del disco para su unidad principal, reemplace su arranque con la ruta completa a la ID. (NOTA: el `/dev/sda` original está comentado y solo para su referencia, no es necesario que lo guarde).

```
#boot = /dev/sda
boot = /dev/disk/by-id/ata-Crucial_CT480M500SSD1_13440956798B
```

Una vez que haya configurado la parte de arranque, puede moverse a las particiones raíz para sus sistemas operativos.

LABEL and UUID

Para usar `label` y `UUID`, debe usar un `initrd`. No funcionará sin él. Ya que `PARTLABEL` en su `lilo.conf` no es compatible, si desea usar una etiqueta allí, tendrá que usar este método. Al crear su `initrd`, debe asegurarse de pasar la opción `-r` para especificar su partición raíz y establecerla en su etiqueta o

UUID. Puede usar el script `/usr/share/mkinitrd/mkinitrd_command_generator.sh`, aunque, si lo está compilando para una versión de kernel que no está ejecutando actualmente (pero tiene instalada), asegúrese de pasar la versión del kernel que pretende utilizar en el script mediante la opción `-k`. A continuación, escupirá un comando `mkinitrd` que puede usar, que solo tendría que ajustar la opción `-r`.

```
root@darkstar:~# /usr/share/mkinitrd/mkinitrd_command_generator.sh -k 4.4.14
#
# mkinitrd_command_generator.sh revision 1.45
#
# This script will now make a recommendation about the command to use
# in case you require an initrd image to boot a kernel that does not
# have support for your storage or root filesystem built in
# (such as the Slackware 'generic' kernels').
# A suitable 'mkinitrd' command will be:

mkinitrd -c -k 4.4.14 -f ext4 -r /dev/sda2 -m jbd2:mbcache:ext4 -u -o
/boot/initrd.gz

root@darkstar:~# blkid
/dev/sda2: UUID="25a4dafe-bbf2-413f-a60c-8c38efc0a122" TYPE="ext4"
PARTUUID="1ae0ebfe-02"
/dev/sda1: UUID="d21b5f69-6d0c-48fb-b67c-912ebb0fd18e" TYPE="swap"
PARTUUID="1ae0ebfe-01"

root@darkstar:~# mkinitrd -c -k 4.4.14 -f ext4 -r "UUID=25a4dafe-bbf2-413f-
a60c-8c38efc0a122" -m jbd2:mbcache:ext4 -u -o /boot/initrd-4.4.14.gz
OK: /lib/modules/4.4.14/kernel/fs/jbd2/jbd2.ko added.
OK: /lib/modules/4.4.14/kernel/fs/mbcache.ko added.
OK: /lib/modules/4.4.14/kernel/fs/jbd2/jbd2.ko added.
OK: /lib/modules/4.4.14/kernel/fs/mbcache.ko added.
OK: /lib/modules/4.4.14/kernel/fs/ext4/ext4.ko added.
35942 blocks
/boot/initrd-4.4.14.gz created.
Be sure to run lilo again if you use it.
root@darkstar:~#
```

Como notará, el comando proporcionado por `mkinitrd` usa `/dev/sda2` como unidad raíz, por lo que necesitamos cambiarlo para que coincida con el UUID de la unidad antes de ejecutar el comando `mkinitrd`. De forma predeterminada, generará el archivo `/boot/initrd.gz`, pero prefiero llamar al mío en función de la versión del kernel (lo cual es útil para garantizar que no sobrescriba un `initrd` en buen estado). Al incluir la partición raíz en su `initrd`, puede eliminarse de `lilo`, ya que no la utilizará (sin embargo, la he dejado pero la he comentado para referencia visual).

Una vez que genere un `initrd`, necesita referenciarlo en la sección. No tengo ningún conjunto de etiquetas para mi unidad principal, así que agregué un ejemplo que está comentado. Puedes usar uno u otro. Como antes, comenté el dispositivo original, pero no tiene que permanecer en el archivo `conf`.



En lugar de cambiar su sección original, puede ser beneficioso mantenerlo allí en caso de que accidentalmente estropee algo y su sistema no arranque, entonces puede



seleccionar su configuración original e ir a arreglar las cosas y volver a intentarlo. Una vez que haya verificado que su sistema se inicia correctamente con su nueva estrofa, si lo desea, puede eliminar la anterior.

```
image = /boot/vmlinuz-generic-3.10.17
initrd = /boot/initrd-3.10.17.gz
#root = /dev/sda2
#root = "UUID=23bce2c2-996d-449e-89cc-0e5029cc6d8d"
#root = "LABEL=Mylabelhere"
label = Slack-generic
read-only

image = /boot/vmlinuz
root = /dev/sda2
label = Slack-default
read-only
```

PARTUUID

Ahora, como se mencionó anteriormente, PARTUUID no requiere un initrd (aunque, funciona bien si lo usa). Sin embargo, lilo tiene la edad suficiente para no tener el soporte adecuado para él, así que hay una solución para que funcione. En lugar de hacer referencia a la raíz como hicimos anteriormente, necesitamos reemplazar la opción de la raíz dentro de un "addappend". Esto agregará algo adicional a la línea de adición inicial en la parte superior de lilo.conf. Tenga en cuenta sus espacios, ya que se colocarán directamente después, por lo que puede ser conveniente incluir un espacio adicional al principio de la línea para asegurarse de que no combine palabras accidentalmente (lo que probablemente causaría un pánico en el núcleo y le evitaría desde el arranque). Para mí, dado que mi partición raíz está en una unidad con un MBR, tendré el PARTUUID acortado en lugar de la longitud normal de 32 caracteres de un PARTUUID adecuado.

```
image = /boot/vmlinuz-generic
addappend = " root=PARTUUID=6f47c81b-02"
label = Slack-partuuid
read-only

image = /boot/vmlinuz-generic
initrd = /boot/initrd-4.4.14.gz
root = "UUID=25a4dafe-bbf2-413f-a60c-8c38efc0a122"
label = Slack-UUID
read-only

image = /boot/vmlinuz-huge
root = /dev/sda2
label = Slack-default
read-only
```



Como se indicó anteriormente, PARTLABEL no es compatible con lilo.

Terminando

Una vez que haya actualizado su archivo `/etc/lilo.conf`, debemos probarlo para asegurarnos de que no haya errores evidentes. Lilo tiene la capacidad de probar una configuración usando la opción `-t`.

```
root@darkstar:~# lilo -t
Warning: LBA32 addressing assumed
Added Slack-partuuid + *
Added Slack-UUID +
Added Slack-default +
The boot sector and the map file have *NOT* been altered.
One warning was issued.
```

La advertencia de LBA32 es normal con `lilo.conf` de Slackware y puede ignorarse (sin embargo, puede eliminarla agregando `LBA32` en la sección Global de su `lilo.conf`). Lo principal que estamos buscando es que todas nuestras estrofas se agregaron y no se emitieron errores. Si eso funcionó, entonces podemos ejecutar lilo por sí mismo para finalizar el proceso.

```
root@darkstar:~# lilo
Warning: LBA32 addressing assumed
Added Slack-partuuid + *
Added Slack-UUID +
Added Slack-default +
One warning was issued.
```

Si llegaste tan lejos, ve y reinicia. Su sistema ya no debería confiar en nombres de dispositivos que puedan cambiar fácilmente.



Lo último para recordar, es que los nombres de sus dispositivos pueden cambiar en cualquier momento. Como siempre debería ser el caso, esté usando o no nombres persistentes, antes de hacer cualquier cosa que pueda borrar una unidad (por ejemplo, `dd`, `fdisk`, etc.), asegúrese de estar usando la unidad correcta nombre. No debes asumir que tu unidad de disco es `/dev/sdc` (porque eso es lo que siempre ha sido) antes de usar `dd` para escribir esa nueva y brillante versión iso de Slackware64-current allí. Si los nombres de sus dispositivos han cambiado, podría estar escribiendo eso en su partición raíz y eliminando todo su sistema operativo. Utilice las herramientas proporcionadas anteriormente (las diversas carpetas en `/dev/disk/`, `blkid` y/o `dmesg`) para verificar qué nombre de dispositivo desea usar antes de ejecutar el comando.

Fuentes

- Originalmente escrito por Jeremy Hansen alias [bassmadrigal](#)
- Traducido por: [Victor](#) 2019/02/05 16:20 (UTC)

[howtos](#), [author bassmadrigal](#)

From: <https://docs.slackware.com/> - **SlackDocs**

Permanent link: https://docs.slackware.com/es:howtos:slackware_admin:how_to_configure_fstab_and_lilo.conf_with_persistent_naming

Last update: **2019/05/11 04:57 (UTC)**

