

Configure SYSLINUX como cargador de arranque en hardware basado en UEFI

SYSLINUX es un cargador de arranque que carga Linux (entre otras cosas) desde el sistema de archivos FAT. [The Syslinux Project](#), del cual forma parte SYSLINUX, el cargador de arranque, también contiene ISOLINUX, EXTLINUX y PXELINUX, básicamente sus variantes para arrancar desde medios ópticos, ext2 / 3 / 4, Btrfs, XFS, UFS / FFS y NTFS o desde red.

El soporte de UEFI parece haberse agregado en la versión 6 de SYSLINUX, mientras que Slackware viene con la versión 4, lo que significa que no se puede configurar durante la instalación de Slackware, a menos que uno prepare los archivos SYSLINUX en otra máquina.

Por qué SYSLINUX

Dicho esto, ¿por qué molestarse con SYSLINUX? Un par de razones:

- Se informa que LILO / ELILO ya no se mantiene, puede que llegue el momento en que tendremos que cambiar a otra cosa.
- El aprendizaje de SYSLINUX puede valer la pena, ya que además de servir como reemplazo de LILO / ELILO, puede hacer muchas otras cosas (por ejemplo, PXE)
- Comparado con GRUB, es, bueno, menos humilde e intimidante. Después de todo, los archivos de configuración se parecen a los de LILO / ELILO.

Por qué UEFI

No es como si hubiera alguna opción; Las máquinas hoy en día son UEFI. Sin embargo, se puede optar por configurar UEFI en el llamado modo "CSM" o, a veces, en "Legacy". En este modo, UEFI actúa como BIOS, es decir, lee el primer sector (MBR) de un disco duro y carga y ejecuta desde allí el cargador de arranque no UEFI, como LILO. A diferencia de UEFI en, digamos, el modo UEFI, cuando lee, carga y ejecuta el cargador de arranque compatible con UEFI desde *partición de arranque EFI*.

Francamente, no hay razones de peso para no usar CSM (tenga en cuenta que es posible usar [GPT partitions without UEFI](#)). Por el contrario, hay razones para evitar el arranque UEFI, porque:

- En algunas máquinas, no es posible desactivar el inicio seguro sin desactivar también el inicio UEFI (es decir, sin activar CSM).
- En algunas máquinas, es posible desactivar el arranque seguro, pero todo lo que venga con la máquina no se iniciará de todos modos.
- En algunas máquinas, UEFI intenta realmente iniciar Windows e ignora o restablece el orden de inicio.
- En algunas máquinas, la instalación de Linux en modo UEFI puede bloquear la máquina.

Pero, al igual que con LILO, puede que llegue el día en que tengamos que seguir adelante.

Instalación

Básicamente, hay dos pasos: primero, preferiblemente incluso antes de instalar Slackware, particione el disco usando [GPT](#) y cree una partición de arranque EFI, luego compile e instale SYSLINUX.

Pero antes de empezar:

- haga una copia de seguridad de sus datos, si tiene algo de valor en cualquiera de las unidades en la máquina
- inicie con la configuración de UEFI (“BIOS”) y verifique que el método de inicio esté establecido en “UEFI”, no en “CMS” o “Legacy”
- y también que el arranque seguro está desactivado

Crear partición de arranque EFI

Es el lugar donde UEFI buscará un cargador de arranque. Se pueden usar varias herramientas para particionar una unidad y crear una partición de arranque EFI,:

- cfdisk (nuevas versiones; viejas hacen solo particiones DOS)
- gdisk, Una herramienta interactiva y contraparte de GPT para fdisk.
- cgdisk, una herramienta basada en menú y GPT equivalente a cfdisk
- sgdisk, versión aparentemente programable de gdisk

Supongamos que tienes 10 GB de disco vacío /dev/sda.

- Por si acaso, borre todas las particiones anteriores de MBR o GPT.

```
# sgdisk -Z /dev/sda
```

- Verifique que la unidad esté limpia.

```
# gdisk -l /dev/sda
GPT fdisk (gdisk) version 1.0.0
```

```
Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present
```

```
Creating new GPT entries.
Disk /dev/sda: 20971520 sectors, 10.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 8C026A41-F188-4521-A72C-D2B826EAA1D1
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 20971486
Partitions will be aligned on 2048-sector boundaries
Total free space is 20971453 sectors (10.0 GiB)
```

Number	Start (sector)	End (sector)	Size	Code	Name
--------	----------------	--------------	------	------	------

- Particione la unidad (utilizando sgdisk o cualquier herramienta mencionada anteriormente). Se crearán tres particiones, una para EFI, una para datos y otra para intercambio.

```
# sgdisk -n 1:0:+512M /dev/sda
# sgdisk -n 2:0:+9G /dev/sda
# sgdisk -n 3 /dev/sda
```

- No te olvides de configurar los códigos de partición correctos. Particiones de arranque EFI tiene código 'ef00'.

```
# sgdisk -t 1:ef00 /dev/sda
# sgdisk -t 2:8300 /dev/sda
# sgdisk -t 3:8200 /dev/sda
```

- Verificar diseño.

```
# gdisk -l /dev/sda
GPT fdisk (gdisk) version 1.0.0
```

Partition table scan:

```
MBR: protective
BSD: not present
APM: not present
GPT: present
```

```
Found valid GPT with protective MBR; using GPT.
Disk /dev/sda: 20971520 sectors, 10.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 3F2101FB-0D6B-481F-98B9-1C6621DB0981
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 20971486
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)
```

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	1050623	512.0 MiB	EF00	
2	1050624	19924991	9.0 GiB	8300	
3	19924992	20971486	511.0 MiB	8200	

- Crear un sistema de archivos FAT32 en la partición de arranque EFI.

```
# mkfs.vfat -F 32 /dev/sda1
```

Eso es más o menos para la partición de arranque EFI. Puede hacer lo que quiera con /dev/sda2 (RAID, LUKS, LVM, cualquier sistema de archivos) e instalar Slackware allí, pero deje la partición EFI sola, esa partición debe estar sin cifrar y el formato FAT32.

Preparar SYSLINUX

Compilación

Descarguelo desde [sources](#) y compile.

```
$ unxz -c syslinux-6.03.tar.xz | tar -x
$ cd syslinux-6.03/
$ make
$ make INSTALLROOT=/some/where install
```

Configurar el cargador de arranque

Ahora copie el cargador de arranque y algunos módulos, así como el kernel y initrd a la partición de arranque EFI creada anteriormente. Supongamos que `/dev/sda1` está montado como `/boot/efi`.

- Copie el cargador de arranque y un módulo para cargar Linux. Eso es lo mínimo. Y vuelva a comprobar que estás copiando archivos efi64.

```
# cd /boot/efi
# mkdir EFI
# mkdir EFI/SYSLINUX
# cd EFI/SYSLINUX
# cp /some/where/usr/share/syslinux/efi64/syslinux.efi .
# cp /some/where/usr/share/syslinux/efi64/ldlinux.e64 .
```

- Copia algunos otros módulos, por ej. para crear un lindo menú de arranque. Para ver la lista completa de módulos y dependencias, consulte [the list](#).

```
# cp /some/where/usr/share/syslinux/efi64/libutil.c32 .
# cp /some/where/usr/share/syslinux/efi64/menu.c32 .
```

- Cree el archivo de configuración `/boot/efi/EFI/SYSLINUX/syslinux.cfg`. A continuación se muestra un ejemplo de menú de arranque simple con dos elementos.
 - `TIMEOUT 100` le dice al cargador de arranque que espere 100 centésimas de segundo, es decir, 10 segundos para la entrada del usuario.
 - Si el usuario no selecciona nada en 10 segundos, lo que esté configurado como `DEFAULT` se pondrá en marcha.
 - La primera entrada del menú es Slackware 14.2..
 - `LABEL` es un nombre que puede ser usado para `DEFAULT`.
 - `MENU LABEL` es una descripción.
 - `LINUX vmlinuz-huge-4.4.14` le dice al cargador de arranque que arranque el kernel `vmlinuz-4.4.14` desde el mismo directorio (es decir, `EFI\SYSLINUX\vmlinuz-4.4.14`)
 - `APPEND /dev/vgroot/V_ROOT 4` es la línea de comando del kernel que le dice al kernel que monte `/dev/vgroot/V_ROOT` como `/` (root). Obviamente, este es un volumen lógico en este caso, pero podría ser, por ejemplo, `/dev/sda2`. El segundo argumento, `'4'`, de hecho pasará de kernel a init, que comenzará a ejecutar el nivel

de ejecución 4 en lugar del predeterminado 3 (según /etc/inittab), es decir, comenzará al modo multiusuario gráfico con el sistema de ventanas X .

- INITRD `initrd.gz-4.4.14` es una imagen `initrd` que se monta como temporal / antes de que se monte / final. Normalmente, se necesita un `initrd` si el kernel tiene que cargar algunos módulos adicionales para acceder al dispositivo raíz, o tiene que realizar algunas acciones para hacerlo accesible (piense en RAID, LUKS, LVM). Si no necesita `initrd`, simplemente elimine esta línea.
- La segunda entrada del menú es prácticamente la misma, excepto que no hay '4' en `APPEND /dev/vgroot/V_ROOT` . En pocas palabras, este es un modo seguro no gráfico. Si el sistema de ventanas X no se inicia por cualquier motivo, puede terminar no solo con la pantalla negra, sino también con el teclado y el mouse bloqueados y, por lo tanto, no puede arreglar la causa. Por lo tanto, es una buena idea tener esta opción preparada de antemano (otro método para usar una máquina inutilizable con una X rota es la entrada SSH, aunque es un método menos confiable)

[/boot/efi/EFI/SYSLINUX/syslinux.cfg](#)

```
UI menu.c32
PROMPT 0
TIMEOUT 100
DEFAULT Slackware 14.2

MENU TITLE Machine Boot Menu

LABEL Slackware 14.2
    MENU LABEL Slackware 14.2
    LINUX vmlinuz-huge-4.4.14
    APPEND /dev/vgroot/V_ROOT 4
    INITRD initrd.gz-4.4.14

LABEL Slackware 14.2 no X
    MENU LABEL Slackware 14.2 no X
    LINUX vmlinuz-huge-4.4.14
    APPEND /dev/vgroot/V_ROOT
    INITRD initrd.gz-4.4.14
```

- Aún en `/boot/efi/EFI/SYSLINUX`, copie el kernel aquí. Huge es la opción más segura. El genérico puede requerir `initrd` y más módulos en él. Si no sabe cuáles son, es posible que la máquina no arranque, así que vaya con la huge.

```
# cp /boot/vmlinuz-huge-4.4.14 .
```

- Cree `initrd` si es necesario y copie aquí también. ejecute `man` para las opciones, pero básicamente el siguiente comando crea un `initrd` que contendrá algunos módulos relacionados con USB para habilitar la entrada del teclado (porque `/dev/sda2` está cifrado (LUKS) y el usuario tendrá que escribir una frase de contraseña).

```
# mkinitrd -c -k 4.4.14 -C /dev/sda2 -L -R -f xfs -m xfs:hid:hid-
generic:usbhid:xhci-hcd:xhci-pci:ehci-hcd:ehci-pci:ohci-hcd:ohci-pci:uhci-
hcd:uhci-pci:usb-storage -r /dev/vgroot/V_ROOT -o /boot/initrd.gz-4.4.14
```

```
# cp /boot/initrd.gz-4.4.14 .
```

- Finalmente, agregue SYSLINUX al menú de arranque de UEFI. Tenga en cuenta que de hecho habrá dos menús de arranque. El primero es el menú de arranque UEFI que normalmente se mostrará solo si presiona alguna tecla (por ejemplo, F12 en Dell, F9 en HP, F11 con la placa base AsRock, a menudo solo Esc) y le presentará una lista de dispositivos o cargadores de arranque para que arranque. Dependiendo de qué tan malo sea tu UEFI, SYSLINUX puede o no aparecer allí.

El segundo menú de inicio es el de SYSLINUX, es decir, lo que escribió en syslinux.cfg y aparecerá si no activa el menú de inicio de UEFI o lo activa y luego selecciona SYSLINUX o dispositivo (disco duro) en el que está activado.

```
# efibootmgr -c -d /dev/sda -p 1 -l \\EFI\\SYSLINUX\\syslinux.efi -L
"SYSLINUX"
Timeout: 2 seconds
BootOrder: 0000,0001,0002
Boot0000* SYSLINUX
Boot0001* UEFI: IP4 Realtek PCIe GBE Family Controller
Boot0002* UEFI: IP6 Realtek PCIe GBE Family Controller
```

- Syslinux se ha agregado con el código 0000 (Boot0000) y es el primero en el orden de arranque. Si no fuera así, habría que cambiar el orden:

```
# efibootmgr -o 0000,0001,0002
```

- En algunas máquinas, puede ser posible cambiar el orden también en UEFI (es decir, su GUI / utilidad de configuración).

Solución de problemas

Si su máquina se queja, no tiene nada que arrancar, eso puede indicar que el comando efibootmgr fue incorrecto o de alguna manera no se registró. Luego, puede intentar forzarlo para que inicie SYSLINUX moviéndolo y cambiándolo de nombre a EFI \ Boot \ bootx64.efi, que es una opción alternativa; La máquina intentará arrancar eso en algún momento.

```
# cd /boot/efi/EFI
# mv SYSLINUX Boot
# cd Boot
# cp syslinux.efi bootx64.efi
```

Sources

- http://www.syslinux.org/wiki/index.php?title=The_Syslinux_Project
- https://wiki.archlinux.org/index.php/Unified_Extensible_Firmware_Interface

[howtos](#), [syslinux](#), [uefi](#)

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

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

Last update: **2019/02/06 04:06 (UTC)**

