EFISTUB

Table of Contents

Le noyau Linux prend en charge le démarrage `EFISTUB` qui permet au micrologiciel EFI de charger le noyau en tant qu'exécutable EFI. L'option peut être activée en définissant `CONFIG_EFI_STUB=y` dans la configuration du noyau.

Avec `EFISTUB`, un noyau peut être démarré directement par une carte mère UEFI ou indirectement en utilisant un chargeur de démarrage. L'utilisation d'un chargeur de démarrage est recommandée lorsqu'on a plusieurs paires de noyau/initramfs et que le menu de démarrage UEFI de la carte mère n'est pas facile à utiliser.

Préparation de la partition ESP

Tout d'abord, il faut créer une partition système EFI et monter la partition pour toutes les options de montage ESP disponibles.

Démarrage avec EFISTUB

le chemin initramfs EFISTUB du noyau Linux doit être relatif à la racine de la partition système EFI et utiliser des barres obliques inverses (conformément aux normes EFI). Par exemple, si les initramfs se trouvent dans `esp/EFI/arch/initramfs-linux.img`, la ligne au format UEFI correspondante doit être `initrd=\EFI\arch\initramfs-linux.img`. Dans les exemples suivants, nous supposerons que tout est sous `esp/`.

UEFI est conçu pour supprimer le besoin d'un chargeur de démarrage intermédiaire tel que GRUB. Si la carte mère a une bonne implémentation UEFI, il est possible d'incorporer les paramètres du noyau dans une entrée de démarrage UEFI pour que la carte mère démarre directement. On peut utiliser **efibootmgr** ou **UEFI Shell v2** pour modifier les entrées de démarrage de la carte mère.

Les implémentations UEFI obsolètes peuvent avoir des problèmes de compatibilité avec

le noyau Linux. S'il existe une version plus récente de l'UEFI de la carte mère avec des corrections de bogues, il faudra la flasher avec l'outil recommandé par le fabricant. Certains firmwares ne transmettent pas les paramètres de ligne de commande des entrées de démarrage de la NVRAM aux fichiers binaires EFI. Dans ce cas, le noyau et les paramètres peuvent être combinés en une image de noyau unifiée.

efibootmgr

Pour créer une entrée de démarrage avec **efibootmgr** qui chargera le noyau:

```
efibootmgr --disk /dev/sdX --part Y --create --label "Arch Linux" --loader
/vmlinuz-linux --unicode 'root=PARTUUID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
rw initrd=\initramfs-linux.img' --verbose
```

ou créer une entrée de démarrage avec **efibootmgr** et hibernation sur la partition de swap:

```
efibootmgr --disk /dev/sdX --part Y --create --label "Arch Linux" --loader
/vmlinuz-linux --unicode 'root=PARTUUID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
resume=PARTUUID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX rw initrd=\initramfs-
linux.img' --verbose
```

Où `/dev/sdX' et` /dev/sdY' sont le numéro de lecteur et de partition où se trouve l'ESP. Modifier les paramètres `root=` et `resume=` pour refléter les partitions racine et swap Linux. S'il est omis, la première partition sur `/dev/sda` est utilisée comme ESP.

l'argument `-u/-unicode` entre guillemets n'est que la liste des paramètres du noyau, il faut donc peut-être ajouter des paramètres supplémentaires (par exemple pour la suspension sur le disque ou le microcode).

Après avoir ajouté l'entrée de démarrage, on peut vérifier que l'entrée a été ajoutée correctement avec:

```
efibootmgr --verbose
```

Pour définir l'ordre de démarrage:

```
efibootmgr --bootorder XXXX,XXXX --verbose
```

Où **XXXX** est le nombre qui apparaît dans la sortie de la commande **efibootmgr** pour chaque entrée.

Il est pratique d'enregistrer la commande pour créer l'entrée de démarrage dans un script shell, ce qui facilite sa modification, par exemple lors de la modification des paramètres du noyau. Pour ce faire, il faut automatiser la suppression des anciennes entrées de démarrage, car **efibootmgr** ne prend actuellement pas en charge la modification des entrées existantes.

efibootmgr avec le fichier .efi

Lorsqu'on a une image **.efi** amorçable, **efibootmgr** peut être utilisé directement pour démarrer le fichier **.efi**:

```
efibootmgr --create --disk /dev/sdX --part partition_number --label "label"
--loader "EFI\folder\file.efi" --verbose
```

Shell UEFI

Certaines implémentations UEFI rendent difficile la modification de la NVRAM à l'aide d'**efibootmgr**. Si **efibootmgr** ne parvient pas à créer une entrée avec succès, on peut utiliser la commande **bcfg** dans **UEFI Shell v2** (c'est-à-dire à partir d'un ISO live), ainsi:

- découvrir le numéro d'appareil où réside votre ESP: `Shell> map`
- utilisr le numéro de périphérique. Pour répertorier le contenu de l'ESP: `Shell> ls fs1`
- afficher les entrées de démarrage actuelles: `Shell> bcfg boot dump`
- ajouter une entrée pour votre noyau: `Shell> bcfg boot add N fs1:\vmlinuz-linux "Arch Linux"` Où N est l'emplacement où l'entrée sera ajoutée dans le menu de démarrage. 0 est le premier élément du menu. Les éléments de menu déjà existants seront déplacés dans le menu sans être supprimés.
- ajouter les options de noyau nécessaires en créant un fichier sur votre ESP: `Shell> edit fs1:\options.txt`
- ajouter la ligne de démarrage. Par exemple: `root=/dev/sda2 ro initrd=\initramfs-linux.img` Ajouter des espaces supplémentaires au début de la ligne dans le fichier. Il y a une marque d'ordre d'octets au début de la ligne qui écrasera tout caractère à côté de lui, ce qui provoquera une erreur lors du démarrage.
- Appuyer sur **F2** pour enregistrer, puis sur **F3** pour quitter.
- Ajouter ces options à votre entrée précédente: `Shell> bcfg boot -opt N fs1:\options.txt`

Répéter ce processus pour toutes les entrées supplémentaires.

Pour supprimer un élément ajouté précédemment, procéder comme suit: `Shell> bcfg boot rm N`

Utilisation d'un script startup.nsh

Certaines implémentations UEFI ne conservent pas les variables EFI entre les démarrages à froid (par exemple VirtualBox avant la version 6.1) et tout ce qui est défini via l'interface du firmware UEFI est

perdu à la mise hors tension.

La spécification Shell UEFI 2.0 établit qu'un script appelé **startup.nsh** à la racine de la partition ESP sera toujours interprété et peut contenir des instructions arbitraires; parmi ceux-ci, on peut définir une ligne de démarrage. Pour cela, monter la partition ESP sur `/boot` et créer un script **startup.nsh** qui contient une ligne de démarrage du noyau. Pour example:

```
vmlinuz-linux rw root=/dev/sdX [rootfs=myfs] [rootflags=myrootflags] \
 [kernel.flag=foo] [mymodule.flag=bar] \
 [initrd=\intel-ucode.img] initrd=\initramfs-linux.img
```

Cette méthode fonctionnera avec presque toutes les versions de firmware UEFI que l'on peut rencontrer dans du matériel réel, on peut l'utiliser en dernier recours. Le script doit être une seule longue ligne. Les sections entre parenthèses sont facultatives et données uniquement à titre indicatif. Les sauts de ligne de style shell sont uniquement destinés à une clarification visuelle. Les systèmes de fichiers FAT utilisent la barre oblique inverse comme séparateur de chemin d'accès et dans ce cas, la barre oblique inverse déclare que l'initramfs est situé à la racine de la partition ESP. Seul le microcode Intel est chargé dans la ligne des paramètres de démarrage; Le microcode AMD est lu à partir du disque plus tard pendant le processus de démarrage; cela se fait automatiquement par le noyau.