Linux kernel options for UEFI and ELILO

The purpose of this article is to inform the user about necessary kernel options for booting from UEFI, and some info about how ELILO and perhaps other EFI bootloaders work, as this is currently difficult to find and understand online.

Required kernel options for UEFI support

The following options are required for booting a kernel from UEFI. These are already set in the stock Slackware kernels.

Booting from UEFI requires that the kernel be the same bitness as the UEFI firmware. This usually means 64-bit, with the exception of new Intel Atom System-on-Chip (2013), older (pre-2008) Apple Macs and Intel Server boards using EFI v1.10 in 32-bit mode, and some Intel Cloverfield ultrabooks. However, you can still enable 32-bit emulation in the kernel and run 32-bit programs in Linux using multilib.

- **Enable the block layer**
  - Partition Types
    - Advanced partition selection
      - EFI GUID Partition support (CONFIG_EFI_PARTITION [=y])
- **Processor type and features**
  - EFI runtime service support (CONFIG_EFI [=y])
  - EFI stub support (CONFIG_EFI_STUB [=y])
  - Build a relocatable kernel (CONFIG_RELOCATABLE [=y])
- **Device Drivers**
  - Graphics support
    - Support for frame buffer devices (CONFIG_FB [=y])
      - EFI-based Framebuffer Support (CONFIG_FB_EFI [=y])
- **File systems**
  - Miscellaneous filesystems
    - EFI Variable filesystem (CONFIG_EFIVAR_FS [=y])

You should **DISABLE** the old efivars sysfs interface found at:

- **Firmware Drivers**
  - EFI (Extensible Firmware Interface) Support
    - EFI Variable Support via sysfs (CONFIG_EFI_VARS [=n])

as it is deprecated in favor of CONFIG_EFIVAR_FS, because it has a 1024-byte variable size limit, and because it can cause data inconsistency issues. However, if you do disable this, you will need a fork of the `efibootmgr` program that supports the new EFI Variable filesystem.

Using the new EFI variable filesystem

In order to use the new EFI variable filesystem interface, you need to remove the old `efibootmgr`

program and replace it with a new one that supports the EFI variable filesystem.

1. Download and install: https://github.com/vathpela/efivar
2. Download and install: https://github.com/rhboot/efibootmgr
3. Run:

```
modprobe efivarfs
mount -t efivarfs efivarfs /sys/firmware/efi/efivars
efibootmgr
```

The EFI System Partition (ESP)

In order to boot from UEFI you need to create an EFI System Partition (ESP) using `gdisk` or `cgdisk` instead of the old `fdisk` and `cfdisk`. The Slackware README_UEFI.TXT recommends a size of 100M, which is also the minimum partition size (except for 4K native drives, where it is 260M). You also need to set the partition hex code to `EF00`, which identifies it as the ESP.

UEFI and ELILO

During the install procedure of Slackware 14.1 for ELILO (the `eliloconfig` script), the following happens and is required for booting from UEFI using ELILO.

1. The EFI System Partition (ESP) is located and mounted. On a running system it is usually found already mounted at `/boot/efi`. This is a special FAT partition that is basically the UEFI firmware replacement of what the MBR was for BIOS. However, it can not only hold a bootloader (all the MBR was capable of), but also config files, the kernel, and other things you might want to access using the UEFI firmware.
2. The three items required by ELILO (and other bootloaders) are copied onto the ESP to `/boot/efi/EFI/Slackware`. These include:
    1. The bootloader. In the case of ELILO it is `elilo.efi`
    2. The config file. In the case of ELILO it is `elilo.conf`
    3. The kernel, usually titled `vmlinuz`
3. A boot entry variable is registered in the UEFI firmware using `efibootmgr`. The exact command that is run is

```
EFI_DEVICE=$(mount | grep vfat | grep -w /boot/efi | cut -b 1-8)
EFI_PARTITION=$(mount | grep vfat | grep -w /boot/efi | cut -f 1 -d ' '
| cut -b 9-)
efibootmgr -q -c -d $EFI_DEVICE -p $EFI_PARTITION -l
"\\EFI\\Slackware\\elilo.efi" -L "Slackware"
```

so if `mount` outputs

```
/dev/sda1 on /boot/efi type vfat (rw)
```

EFI_DEVICE would be `/dev/sda` and EFI_PARTITION would be `1`, each components of

`/dev/sda1`, which is the ESP.

If for some reason you cannot register a boot entry with the UEFI firmware, you should put the bootloader at `/EFI/boot/bootx64.efi`. This special location allows the UEFI firmware to run the bootloader without any boot entry.

Upgrading your kernel

This task is now much easier than it used to be. All you really need to do is copy `vmlinuz` onto the ESP on top of the old kernel. No need to edit any configs or add any boot entries, unless you want to. Note that you can edit the config in place and ELILO will pick up the changes on next boot, no need to run any commands like with the old lilo.

Updating your UEFI firmware

As all UEFI firmware has a flashing utility built-in, it is now much easier to update the firmware. All you have to do is copy the new firmware onto the ESP and the UEFI firmware should recognize it when you go to the flashing utility menu. However, remember that flashing the firmware can still potentially brick the system, especially if it is interrupted during the flashing process.

Updating your UEFI firmware may reset your settings and prevent you from booting unless you plan ahead.

My UEFI settings were reset and I can't boot, or planning ahead to avoid surprises

This can happen either after updating the UEFI firmware or after replacing the CMOS battery. There are three main ways to fix it:

1. Probably the easiest and most convenient way is to use the default boot location. Note that some UEFI firmwares do NOT support the default boot location, so this will not work. However, if it does, you won't have to worry about the system not booting again. To do this you can boot into the Slackware install DVD, mount the ESP and copy the files to the following places:
    1. `elilo.efi` → `/EFI/boot/bootx64.efi`
    2. `elilo.conf` → `/EFI/boot/elilo.conf`
    3. `vmlinuz` → `/EFI/boot/vmlinuz`
2. You can boot into the Slackware install DVD, run through the menus, and reinstall elilo.
3. You can download and install one of the following EFI shells to the root directory of your ESP (that is `/` NOT `/EFI`).
    1. This version supports only UEFI version 2 and up
    2. This version may support earlier UEFI versions
    3. Boot into the shell and run:

```
bcfg boot add 0 fs0:\EFI\Slackware\elilo.efi Slackware
```

External Links

A comprehensive analysis of the ESP and default boot behavior.

Sources

- Original author: metaschima
- https://wiki.archlinux.org/index.php/UEFI
- https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/plain/Documentation/x86/x86_64/uefi.txt
- `eliloconfig` bash script by Patrick Volkerding

howtos, author metaschima