



从开机键到 **Linux** 桌面的奇幻之旅

Sharelter

10/27/24

自我介绍

- AOSC 打包工
- Github: *@Sharelter*

点亮!

- BIOS (Basic Input/Output System) 或是 UEFI (Unified Extensible Firmware Interface) 程序会在按下电源键之后马上介入, 负责初始化硬件设备, 加载引导程序。
- ta 会检测各种硬件设备, 检查是否满足启动条件 不然就摆烂然后哭给你看 (发出嘀嘀嘀的声音)。
- 不过, 系统完全启动之后, 这些设备中的大多数都会被驱动程序接管。
- (所以在一些比较奇怪的硬件上, 会出现 UEFI 阶段没有显示输出的情况。)

引导!

- 接下来，BIOS/UEFI 会根据系统配置选择启动设备。
- BIOS 系统的 boot loader 位于引导设备的第一个扇区：MBR。它占据磁盘的前 512 个字节。
- 另一方面，UEFI 系统将所有启动数据存储在 .efi 文件中。这些文件位于 EFI 系统分区中，包含 boot loader

- 可以用 fdisk 检查磁盘上的分区们:

```
fdisk -l
```

```
Disk /dev/nvme0n1: 232.91 GiB, 250059350016 bytes, 488397168 sectors
```

```
Disk model: Samsung SSD 960 EVO 250GB
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: gpt
```

```
Disk identifier: 6907D1B3-B3AB-7E43-AD20-0707A656A1B5
```

Device	Start	End	Sectors	Size	Type
/dev/nvme0n1p1	2048	1050623	1048576	512M	EFI System
/dev/nvme0n1p2	1050624	34605055	33554432	16G	Linux swap
/dev/nvme0n1p3	34605056	488397134	453792079	216.4G	Linux filesystem

加载!

- Boot loader 是一个加载操作系统的小程序，它的主要工作是在磁盘上找到内核 (kernel)，插入内存中，根据指定选项执行 kernel。
- 作为一个开放的系统，Linux 有非常多的 loader 可供选择：

大概有这么多 (在 ArchWiki 偷了张图过来)

Name ↕	Firmware		Partition table		Multi-boot ↕	File systems ↕	Notes ↕
	BIOS ↕	UEFI ↕	MBR ↕	GPT ↕			
EFI boot stub	-	Yes ¹	Yes	Yes	-	Inherited from firmware ²	The kernel is a valid EFI executable which can be directly launched from UEFI or from another UEFI boot loader.
Unified kernel image	-	Yes ³	Yes	Yes	-	Inherited from firmware ²	systemd-stub(7) , a kernel, initramfs and kernel command line packed into EFI executable to be loaded directly from UEFI firmware or another boot loader.
GRUB	Yes	Yes ³	Yes	Yes	Yes	Built-in	Supports RAID, LUKS (but not Argon2 PBKDFs) and LVM (but not thin provisioned volumes). See GRUB for setup-specific limitations.
Limine	Yes	Yes	Yes	Yes	Yes	Limited	
rEFInd	No	Yes	Yes	Yes	Yes ⁴	Extensible ^{2,5}	Supports auto-detecting kernels and parameters without explicit configuration, and supports fastboot [2] ⓘ.
Sylinux	Yes	Partial ¹	Yes	Yes	Partial	Limited	No support for certain file system features. Can only access the file system it was installed to .
systemd-boot	No	Yes ³	Manual ⓘ	Yes	Yes ⁴	Extensible ^{2,5}	Can only launch binaries from the ESP it is installed to or from the Extended Boot Loader Partition (XBOOTLDR partition) on the same disk. Automatically detects unified kernel images placed in <code>esp/EFI/Linux/</code> .
GRUB Legacy	Yes	No	Yes	No	Yes	Limited	Discontinued ⓘ in favor of GRUB .
LILO	Yes	No	Yes	Partial ⓘ	Yes	Limited	Discontinued ⓘ due to limitations (e.g. with Btrfs, GPT, RAID, encryption).

加载!

- 尽管有这么多选择，但大部分（非嵌入式）现代 Linux 发行版都使用 GRUB（GRand Unified Boot Loader），因为它的功能非常丰富：
 - 能启动多个操作系统
 - 同时启动图形和文本界面
 - 可通过 serial console 轻松使用
 - 用于交互式配置的强大命令行界面
 - 基于网络的无磁盘引导

加载!

- 现在，让我们再来看看 GRUB2 是怎么 work 的：
 - 在启动时接替 BIOS 或 UEFI
 - 加载自己
 - 把 Linux kernel dump 进内存
 - 将执行权交给内核

加载!

- GRUB 配置文件默认位于 默认位于 /boot/grub

```
sharelter@hplaptop:~  
● > ls -l /boot/grub  
total 44  
drwxr-xr-x 2 root root 4096 Apr  3 2023 fonts  
-rwxr-xr-x 1 root root 5948 Jun  4 16:52 grub.cfg  
-rwxr-xr-x 1 root root 1024 Apr  3 2023 grubenv  
drwxr-xr-x 2 root root 4096 Apr  3 2023 locale  
drwxr-xr-x 3 root root 4096 Apr  3 2023 themes  
drwxr-xr-x 2 root root 20480 Apr  3 2023 x86_64-efi
```

加载!

- 值得注意的是，这里的配置文件是由 grub-mkconfig 自动生成的

```
sharelter@hplaptop:~
```

```
● > cat /boot/grub/grub.cfg
```

```
#
```

```
# DO NOT EDIT THIS FILE
```

```
#
```

```
# It is automatically generated by grub-mkconfig using templates
```

```
# from /etc/grub.d and settings from /etc/default/grub
```

```
#
```

内核!

- 内核在系统的低层次（*kernel space*）发挥作用，在机器硬件和程序之间进行交互。
- 使用 boot loader 启动的内核是一个 vmlinux image ，
- 在这里它会遵循预定义的 procedure：
 - 就地自解压
 - 执行硬件检查
 - 访问重要的外围硬件
 - 运行 init 进程

Init !

- Init daemon 是内核启动的第一个进程，所以它的进程号 (PID)=1，负责启动操作系统运行所需的其他进程。
- 同样有相当多的可选项：
 - Systemd
 - SysVinit
 - OpenRC
 - S6
 - etc...

Systemd

- Systemd 执行一系列任务：
 - 探测所有剩余硬件
 - 挂载文件系统
 - 启动和终止各种服务
- Systemd 使用 `/etc/systemd/system/default.target` 文件来决定 Linux 系统启动时的状态或目标。

[*] started sddm.service

- 好耶，熟悉的登录界面出现啦 ~

10:31 AM

Tuesday, April 4, 2023



Live-CD User



Sleep



Restart



Shut Down



Other...

Q&A



Thanks!

