

用于Linux的开放BIOS开源项目介绍 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E7_94_A8_E4_BA_8ELinu_c103_144166.htm 在很多系统上，引导时间中有很大一部分都花费在为 MS-DOS 提供传统支持上面了。有很多项目，包括 LinuxBIOS 和 Open Firmware，都试图使用最新的代码来替换原有的 BIOS 系统，它们只实现加载并运行 Linux 内核所必需的功能。本文对这个领域的内容简要进行了介绍。响铃！尽管在 PC 硬件加电时喇叭响一下看起来似乎是件非常自然的事情，但实际上却有一些代码来驱动喇叭发声。这段代码就是引导固件。在大部分 PC 上，这都称为 BIOS（这个单词是 basic input/output system（基本输入/输出系统）的缩写）。BIOS 提供了底层的硬件支持，早期的 x86 操作系统就是使用它们来访问磁盘、显示器和其他东西的。BIOS 要做的第一件事情是执行各种加电测试：确定（还可能要测试）可用内存、确定时钟速度等。如果测试成功，机器的喇叭就会响一声。这个过程就称为加电测试（power-on self test）或 POST.计算机对于自己是相当幽默的，这个术语通常会被当作动词来使用：“这台机器根本通不过 POST 测试，因此我们应该更换内存”。通常的诊断包括响铃代码（不同供应商提供的代码都不相同），或者可以写入到某个特定裸地址的代码。有些后插的卡可以简单地访问这些代码；标准的解决方案是诊断代码都写到 80 端口中。有些制造商会销售一种卡，它可以以 16 进制的形式显示最新写入到 80 端口的内容。如果我们要进行严格调试，可能就会希望有一个这种卡，或者希望使用一个更好的发明，例如 PC Weasel，它可以

记录最新的 (256) POST 代码以供大家阅读。(有关 PC Weasel 的更多信息请参看下面的参考资料一节的内容。) 当然, 这些代码的确切含义对于各个 BIOS 来说不尽相同, 只有部分供应商提供了文档。幸运的是, 开源供应商提供了很好的文档。BIOS 还为我们实现了哪些功能? 诸如 MS-DOS 之类的操作系统可以加载其他设备驱动程序, 例如 CD-ROM 驱动器, 但是需要所有硬件驱动程序在启动时就已经加载上来了。为这些驱动程序提供的标准接口是由 BIOS 来处理的, 正是由于这个原因, BIOS 需要对设备进行探测、识别, 还可能要进行初始化。同样, BIOS 要负责对内存进行初始化。并非所有的操作系统都需要对内存进行初始化, 但是早期的 DOS 系统通常都需要进行这种操作, 即使在今天大部分 BIOS 为了兼容性的目的也需要进行初始化。这个过程自己可能需要很长时间才能完成, 很多现代系统允许彻底或部分地将其禁用。同时, BIOS 还会试图确定系统中有多少内存可用。其他的引导时操作还可能包括对处理器缓存的初始化和启用, 配置双 CPU, 构建有关处理器的信息表, 构建连接到系统中的 PCI 设备, 甚至运行这些设备提供的引导 ROM, 这可以加载其他驱动程序。这里有很多工作需要做。实际上这些工作是如此繁杂, 我的某些系统要花一分钟甚至更长时间才能完成 POST 和后续的驱动程序初始化过程。BIOS 可以执行各种硬件扫描, 从而寻找可引导的设备, 在某些系统上, BIOS 甚至可以执行通过以太网进行的网络引导。我的一个系统就会花大约 5 秒钟的时间来初始化网络引导参数, 即使在禁用了网络引导功能的情况下也是如此。这可真令人懊恼! 最后一点 (但不是最不重要的一点), BIOS 还需要做相当多的初始化

工作。不管您要引导什么系统，这些工作中的一部分（但并非全部）都会非常有用。为设备分配中断请求（IRQ）的确是一个非常有用的服务，因为它允许 OS 只获取一个服务列表并开始运行，而不用对它们进行编程。很多设备都有配置寄存器，BIOS 可以基于系统可写内存中的设置向其中写入合理或正确的值。（通常来说，这种内存都称为 CMOS，尽管它们并非严格要求使用这种技术来实现。）在 BIOS 完成上面这些功能之后又会发生什么呢？它会在某个地方（通常是在磁盘上）查找一段代码，并运行它，这通常会加载一个操作系统。如果操作系统是 DOS，或其他类似的东西，那么这种设置工作就意味着我们可以很快就有自己的命令提示符了。双重努力但是对于 Linux、BSD 或 Windows？来说，操作系统有自己的驱动程序。因此，操作系统可以丢弃连接到系统上的 PCI 设备列表，并开始加载这些设备。BIOS 所做的工作大部分会被忽略掉，操作系统一旦加载 SCSI 驱动程序之后，就会自行扫描 SCSI 总线。BIOS 并没有做什么其他工作，仅仅是提供信息而已，并且它所提供的很多信息也都不会被使用。BIOS 实际上需要做的事情是加载第一段代码（称为引导加载程序（bootstrap loader）或 bootloader），并让机器开始运行。因此，在负载相当重的机器上，我们需要等待每个设备都被探测两次。通常，SCSI 控制器的引导 ROM 要花费相当长的时间来扫描设备。更为糟糕的是，在引导 ROM 运行时，其他东西都无法运行。与之不同的是，现代操作系统可以很好地加载 SCSI 驱动程序，发起一个总线重置命令，在检查设备完成之前就可以继续加载其他驱动程序，并执行网络设备的加载。简而言之，操作系统的扫描要比 BIOS 的扫描

速度更快。确切的速度差异取决于操作系统。不过我们知道 Linux 的引导时扫描可以非常快。这意味着 BIOS 不仅要花从加电到最后一个内核模块加载时间的一半在这上面；有时还可能会花费更多时间。当重要性较低的内核模块（例如声卡驱动程序）在引导过程最后加载时（此时一些重要的东西，例如 Web 服务器，可能已经加载了），这种现象更加明显。我们希望要的是一种不用等待 BIOS 来做大量有关内核的工作就可以加载内核的方法 这样可以更快、更好、更可靠地完成引导过程。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com