分类： 理解计算机

上一篇：纪录片《TPB AFK

下一篇：玉门人家照相馆

计算机是如何启动的？

作者： 阮一峰

日期： 2013年2月16日

从打开电源到开始操作，计算机的启动是一个非常复杂的过程。

我一直搞不清楚，这个过程到底是怎么回事，只看见屏幕快速滚动各种提示…… 这几天，我查了一些资料，试图搞懂它。下面就是我整理的笔记。

零、boot的含义

先问一个问题，"启动"用英语怎么说？

回答是boot。可是，boot原来的意思是靴子，"启动"与靴子有什么关系呢？ 原来，这里的boot是bootstrap（鞋带）的缩写，它来自一句谚语：

```
"pull oneself up by one's bootstraps"
```

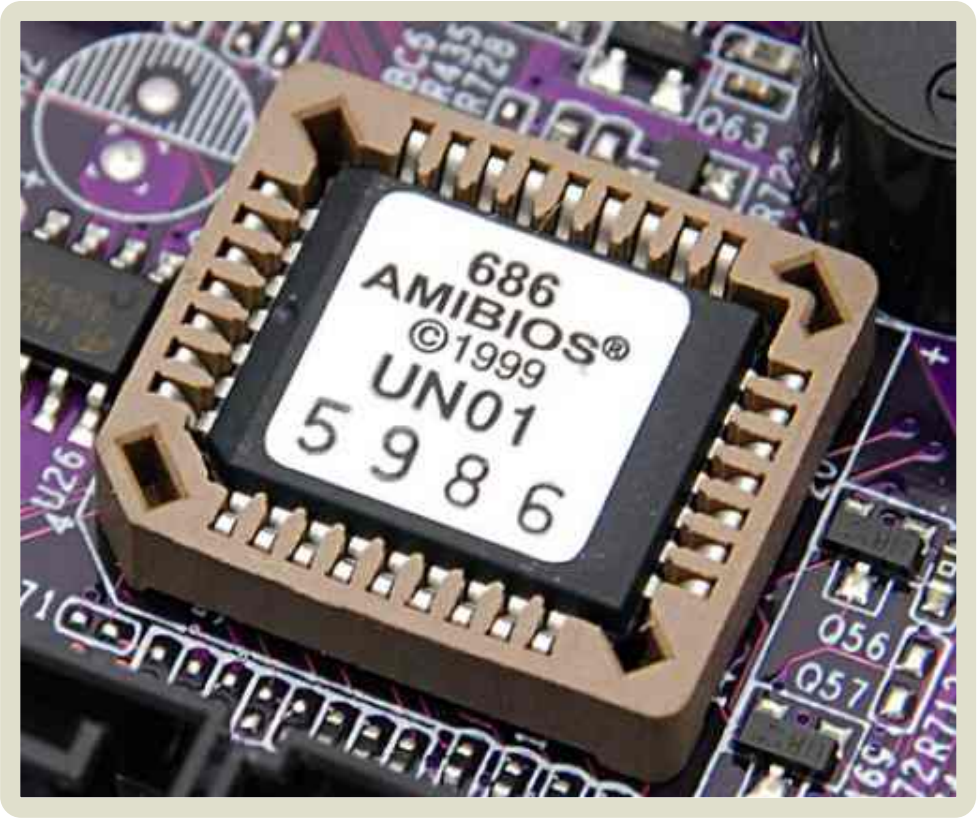
字面意思是"拽着鞋带把自己拉起来"，这当然是不可能的事情。最早的时候，工程师们用它来比喻，计算机启动是一个很矛盾的过程：必须先运行程序，然后计算机才能启动，但是计算机不启动就无法运行程序！

早期真的是这样，必须想尽各种办法，把一小段程序装进内存，然后计算机才能正常运行。所以，工程师们把这个过程叫做"拉鞋带"，久而久之就简称为boot了。

计算机的整个启动过程分成四个阶段。

一、第一阶段：BIOS

上个世纪70年代初，"只读内存"（read-only memory，缩写为ROM）发明，开机程序被刷入ROM芯片，计算机通电后，第一件事就是读取它。

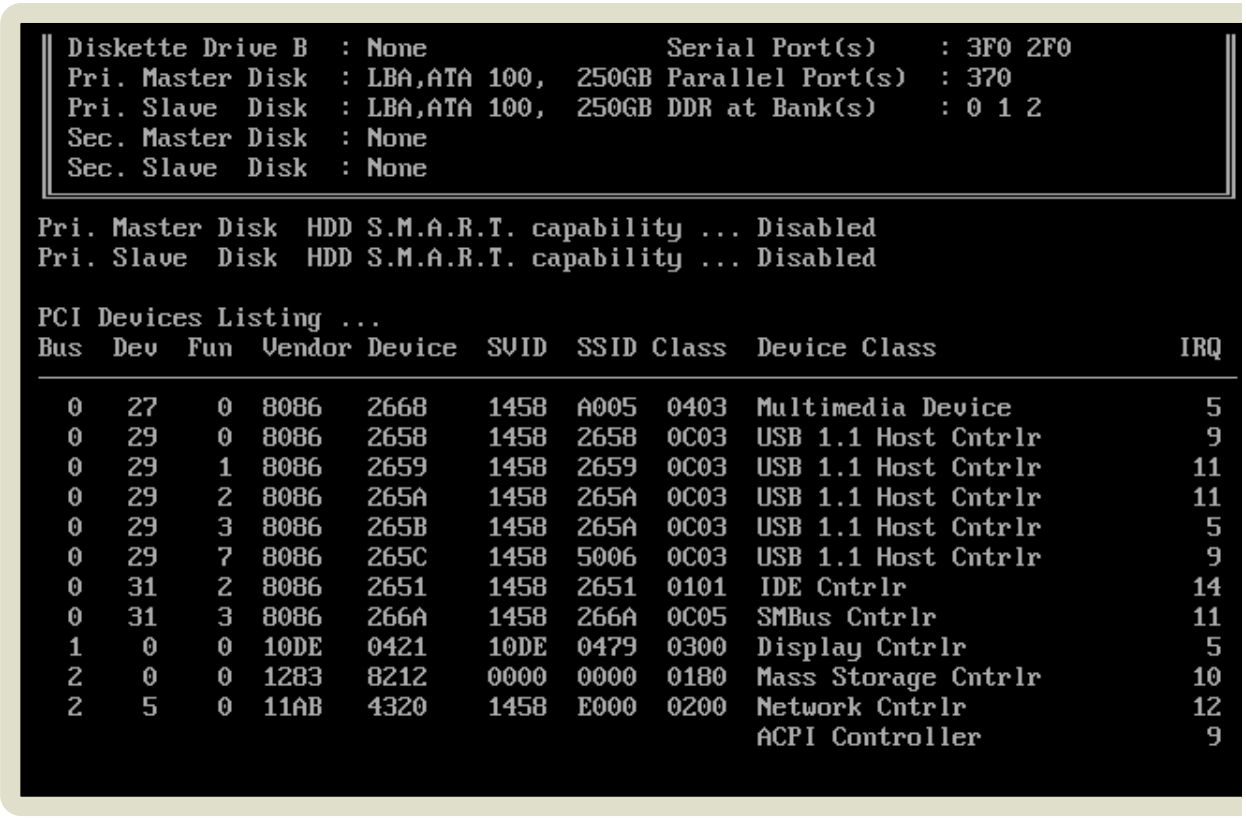


这块芯片里的程序叫做"基本輸出輸入系統"（Basic Input/Output System），简称为BIOS。

1.1 硬件自检

BIOS程序首先检查，计算机硬件能否满足运行的基本条件，这叫做"硬件自检"（Power-On Self-Test），缩写为POST。

如果硬件出现问题，主板会发出不同含义的蜂鸣，启动中止。如果没有问题，屏幕就会显示出CPU、内存、硬盘等信息。

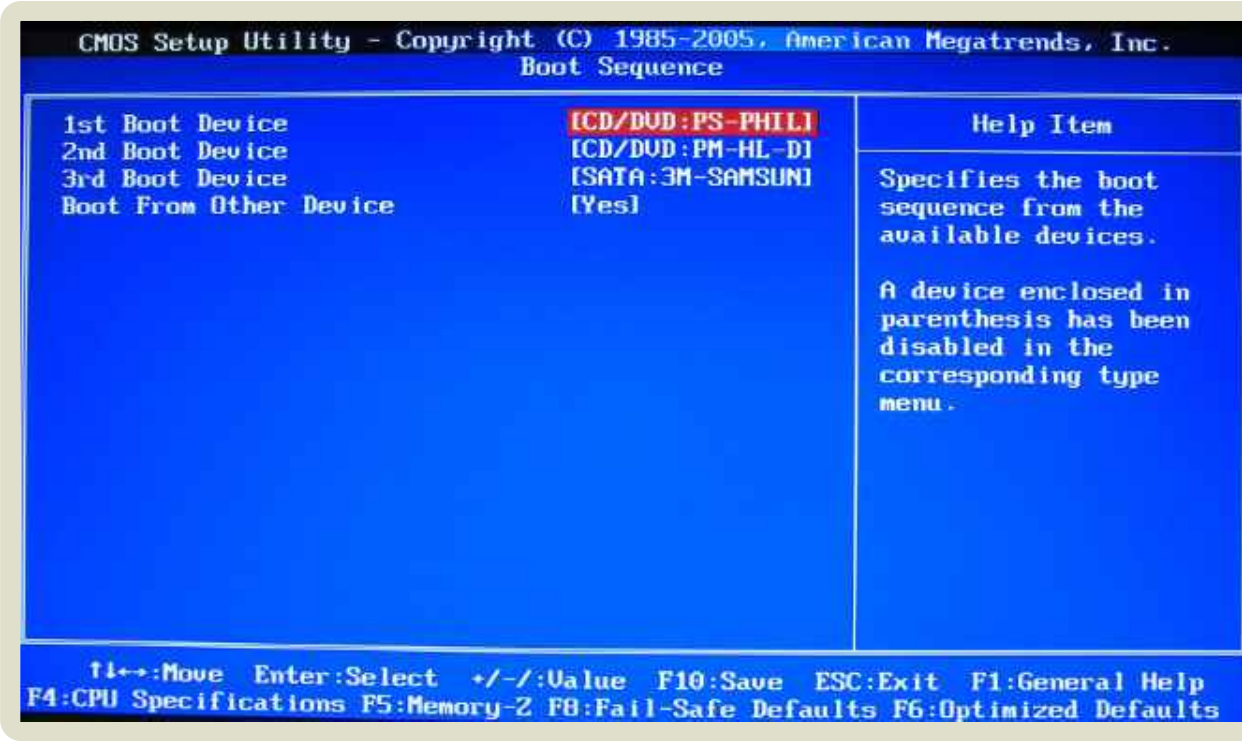


1.2 启动顺序

硬件自检完成后，BIOS把控制权转交给下一阶段的启动程序。

这时，BIOS需要知道，"下一阶段的启动程序"具体存放在哪一个设备。也就是说，BIOS需要有一个外部储存设备的排序，排在前面的设备就是优先转交控制权的设备。这种排序叫做"启动顺序"（Boot Sequence）。

打开BIOS的操作界面，里面有一项就是"设定启动顺序"。



二、第二阶段：主引导记录

BIOS按照"启动顺序"，把控制权转交给排在第一位的储存设备。

这时，计算机读取该设备的第一个扇区，也就是读取最前面的512个字节。如果这512个字节的最后两个字节是0x55和0xAA，表明这个设备可以用于启动；如果不是，表明设备不能用于启动，控制权于是被转交给"启动顺序"中的下一个设备。

这最前面的512个字节，就叫做"主引导记录"（Master boot record，缩写为MBR）。

2.1 主引导记录的结构

"主引导记录"只有512个字节，放不了太多东西。它的主要作用是，告诉计算机到硬盘的哪一个位置去找操作系统。

主引导记录由三个部分组成：

（1） 第1-446字节：调用操作系统的机器码。

（2） 第447-510字节：分区表（Partition table）。

（3） 第511-512字节：主引导记录签名（0x55和0xAA）。

其中，第二部分"分区表"的作用，是将硬盘分成若干个区。

2.2 分区表

硬盘分区有很多好处。考虑到每个区可以安装不同的操作系统，"主引导记录"因此必须知道将控制权转交给哪个区。

分区表的长度只有64个字节，里面又分成四项，每项16个字节。所以，一个硬盘最多只能分四个一级分区，又叫做"主分区"。

每个主分区的16个字节，由6个部分组成：

（1） 第1个字节：如果为0x80，就表示该主分区是激活分区，控制权要转交给这个分区。四个主分区里面只能有一个是激活的。

（2） 第2-4个字节：主分区第一个扇区的物理位置（柱面、磁头、扇区号等等）。

（3） 第5个字节：主分区类型。

（4） 第6-8个字节：主分区最后一个扇区的物理位置。

（5） 第9-12字节：该主分区第一个扇区的逻辑地址。

（6） 第13-16字节：主分区的扇区总数。

最后的四个字节（"主分区的扇区总数"），决定了这个主分区的长度。也就是说，一个主分区的扇区总数最多不超过2的32次方。

如果每个扇区为512个字节，就意味着单个分区最大不超过2TB。再考虑到扇区的逻辑地址也是32位，所以单个硬盘可利用的空间最大也不超过2TB。如果想使用更大的硬盘，只

有2个方法：一是提高每个扇区的字节数，二是增加扇区总数。

三、第三阶段：硬盘启动

这时，计算机的控制权就要转交给硬盘的某个分区了，这里又分成三种情况。

3.1 情况A：卷引导记录

上一节提到，四个主分区里面，只有一个是激活的。计算机会读取激活分区的第一个扇区，叫做"卷引导记录"（Volume boot record，缩写为VBR）。

"卷引导记录"的主要作用是，告诉计算机，操作系统在这个分区里的位置。然后，计算机就会加载操作系统了。

3.2 情况B：扩展分区和逻辑分区

随着硬盘越来越大，四个主分区已经不够了，需要更多的分区。但是，分区表只有四项，因此规定有且仅有一个区可以被定义成"扩展分区"（Extended partition）。

所谓"扩展分区"，就是指这个区里面又分成多个区。这种分区里面的分区，就叫做"逻辑分区"（logical partition）。

计算机先读取扩展分区的第一个扇区，叫做"扩展引导记录"（Extended boot record，缩写为EBR）。它里面也包含一张64字节的分区表，但是最多只有两项（也就是两个逻辑分区）。

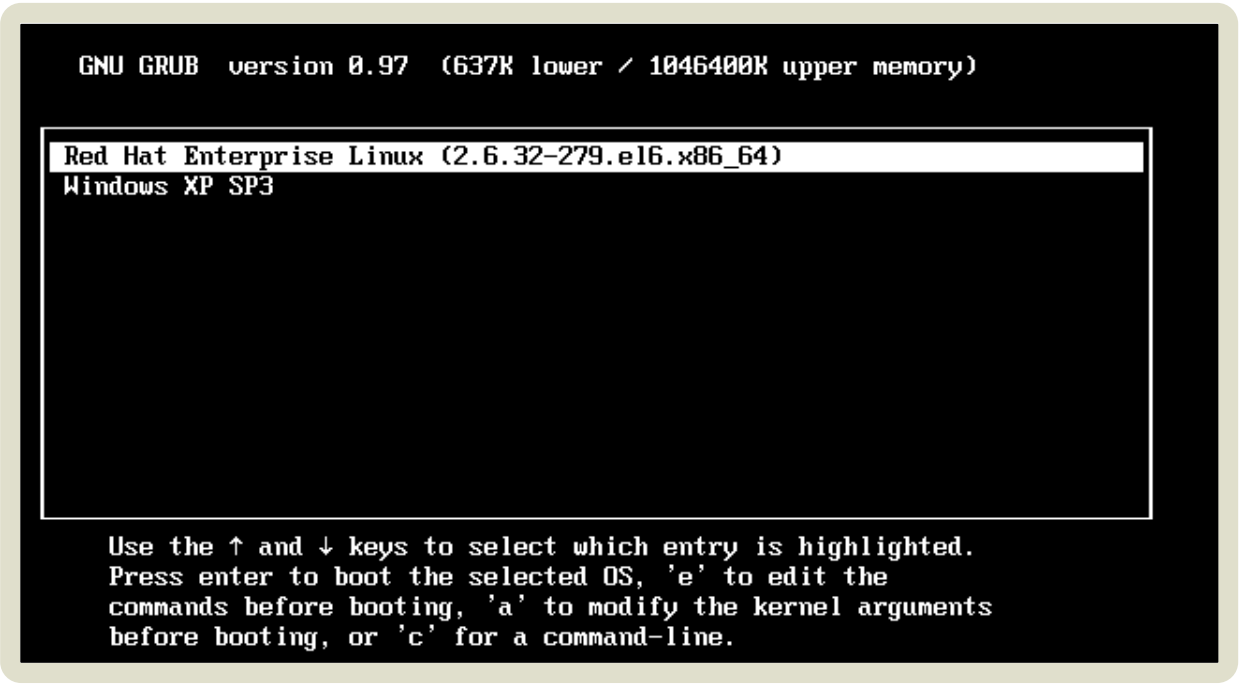
计算机接着读取第二个逻辑分区的第一个扇区，再从里面的分区表中找到第三个逻辑分区的位置，以此类推，直到某个逻辑分区的分区表只包含它自身为止（即只有一个分区项）。因此，扩展分区可以包含无数个逻辑分区。

但是，似乎很少通过这种方式启动操作系统。如果操作系统确实安装在扩展分区，一般采用下一种方式启动。

3.3 情况C：启动管理器

在这种情况下，计算机读取"主引导记录"前面446字节的机器码之后，不再把控制权转交给某一个分区，而是运行事先安装的"启动管理器"（boot loader），由用户选择启动哪一个操作系统。

Linux环境中，目前最流行的启动管理器是Grub。



四、第四阶段：操作系统

控制权转交给操作系统后，操作系统的内核首先被载入内存。

以Linux系统为例，先载入/boot目录下面的kernel。内核加载成功后，第一个运行的程序是/sbin/init。它根据配置文件（Debian系统是/etc/initab）产生init进程。这是Linux启动后的第一个进程，pid进程编号为1，其他进程都是它的后代。

然后，init线程加载系统的各个模块，比如窗口程序和网络程序，直至执行/bin/login程序，跳出登录界面，等待用户输入用户名和密码。

至此，全部启动过程完成。

（完）

文档信息

- 版权声明：自由转载-非商用-非衍生-保持署名（创意共享3.0许可证）
- 发表日期： 2013年2月16日
- 更多内容： 档案 » 理解计算机

- 购买文集： 《如何变得有思想》
- 社交媒体： twitter , weibo
- Feed订阅：

相关文章

- **2014.11.11:** 编译器的工作过程

  > 源码要运行，必须先转成二进制的机器码。这是编译器的任务。

- **2014.09.07:** 数据压缩与信息熵

  > 1992年，美国佐治亚州的WEB Technology公司，宣布做出了重大的技术突破。

- **2014.07.04:** 数据库的最简单实现

  > 所有应用软件之中，数据库可能是最复杂的。

- **2013.11.29:** Stack的三种含义

  > 学习编程的时候，经常会看到stack这个词，它的中文名字叫做"栈"。

广告（购买广告位）

留言（71条）

海纳百川 说：

沙发，学习了。虽是计算机专业，但对底层的东西了解的不深，往往又拿透明的来搪塞自己。

2013年2月16日 14:21 | 档案 | 引用

lvzongting 说：

bios不一定存在，bios只是提供了一个固定的方式，去访问其他输入输出设备，存在的目的是可以将硬件的变化简化，这样内核就可以通过和固定的bios通信来获取基本的硬

件地址信息，然后再获得更充分的硬件地址信息。没有bios的计算机，就没有这么一个固定的方式，随着硬件的改变，就要静态的改变内核内部设备的入口地址，然后重新编译。可以这么说bios是使得现在操作系统变得通用的一个利器。比如现在不同的arm手机的内核不能随便换着用，而有bios的计算机内核即使硬件不那么一样，却可以用同样的内核。还比如现在很多时候都会将硬件地址的map独立于内核写到一个单独的文件中，内核载入的时候会从这些文件中读取硬件的入口地址。
从上电开始，处理器的指令计数器会被初始化成一个值(这个值不一定是0x0000H，因为一般还会在前面放中断向量表)，然后从这个地址开始执行程序。这些程序有可能放到片内的flash，有可能放到片外的flash，或是RAM里面。由于RAM是掉电易失的，所以可能需要先从其他存储设备中载入内存然后再从内存中调入片内进行运算。

2013年2月16日 14:40 | 档案 | 引用

snowqiang 说：

这样来看， boot翻译成'自举'还是挺形象的。

2013年2月16日 15:53 | 档案 | 引用

Wht'for 说：

>>>计算机通电后，第一件事就是读取它。
我更好奇，是*什么*又是*怎么*读取的。

2013年2月16日 17:28 | 档案 | 引用

张刚 说：

MBR是个很重要的概念，在经历N次启动硬盘找不到OS的情况才能深刻了解。
VBR（卷引导记录）应该是硬盘在逻辑卷（LVM）管理方式的情况，不过LVM可以是管理多个硬盘，或者是单个硬盘的某个分区。

2013年2月16日 18:40 | 档案 | 引用

Mike 说：

For UEFI, it is a different story.

2013年2月16日 18:41 | 档案 | 引用

Canson 说：

> 引用**Mike**的发言：
>
> For UEFI, it is a different story.

的确，或者说即使不是UEFI，Legacy的Bios的故事也有别的变种。BIOS 工程师路过

2013年2月16日 19:06 | 档案 | 引用

OJ77 说：

很不错，解决了很多疑惑！

2013年2月16日 20:42 | 档案 | 引用

大磊童鞋 说：

我曾经也写过一个类似的启动过程，更加注重的是linux/unix系统，如果从加电到grub1/2的执行，分别考虑到了EFI和BIOS的过程，地址是：http://www.dalei.org/linux-unix-boot-process

2013年2月16日 20:46 | 档案 | 引用

iyue 说：

boot这个说明写的真好

2013年2月16日 21:44 | 档案 | 引用

cumirror 说：

谢谢阮兄的好文。

下面这篇文章中有关于硬盘分区的详细介绍：
http://wenku.baidu.com/view/4ec485d4b9f3f90f76c61bc0.html

2013年2月16日 22:44 | 档案 | 引用

goddyang 说：

我印象中MBR和partition table不是包含关系，MBR只包含最前面的446个字节的内容

2013年2月17日 00:15 | 档案 | 引用

stone 说：

> 引用**goddyang**的发言：
>
> > 我印象中MBR和partition table不是包含关系，MBR只包含最前面的446个字节的内容

有几次安装系统，分区表正常，但是不能引导，重写MBR后可以引导了，所以我也有这种感觉。

2013年2月17日 10:01 | 档案 | 引用

夜弓 说：

Gpt分区的还是和以上过程有些差别

2013年2月17日 10:37 | 档案 | 引用

姐姐 说：

Master boot record是Master boot record , Partition table是Partition table，两者没有包含关系吧……

2013年2月17日 11:37 | 档案 | 引用

skywalker1114 说：

阮兄：Bootstrap不是鞋带的意思，应该是“鞋子背带”的意思（http://en.wiktionary.org/wiki/bootstrap）

在这里隐喻表示一种不需要外部帮助自己能够处理事情的情形。
“pull oneself up by one's bootstraps”最初来自于《The Surprising Adventures of Baron Munchausen》这本书里的一个故事：主人公Baron Munchausen不小心掉进了一片沼泽，他通过自己的bootstraps将自己拉了出来（当然有童话神奇的色彩）。事实上在19世纪初美国就有"pull oneself over a fence by one's bootstraps"的语言，意思是“做荒谬不可能完成的事情”。
参考：http://en.wikipedia.org/wiki/Bootstrapping

2013年2月17日 15:14 | 档案 | 引用

endle 说：

> 引用夜弓的发言：
>
> Gpt分区的还是和以上过程有些差别

请问您能讲一下区别吗？我记得GPT是可以支持大于2T的硬盘空间的吧

2013年2月17日 18:09 | 档案 | 引用

bj0629 说：

从逻辑和原理上来分析，BIOS是为了适应不同的操作系统。如论是DOS，Windows还是UNIX都可以通过读取 BIOS 中硬盘、软盘、键盘、显示器、日期等实现外部I/O调用，以用于实现某种计算机平台的应用。如果只是为了专用的操作系统，没BIOS也照样可以工作。就是通过 reset vector 复位矢量，一个绝对跳转去执行相应的程序。

2013年2月17日 18:57 | 档案 | 引用

dakwing 说：

bios提供类似API的东西（系统中断），如果没有bios，想操作输入输出设备那只能用out和in了，哪怕只是向屏幕输出一个字符或发出beep声

2013年2月17日 20:25 | 档案 | 引用

mk 说：

讲得很深入，

2013年2月18日 10:39 | 档案 | 引用

noodles 说：

单个硬盘最大只能利用2T的空间

这个不是很明白

2013年2月18日 10:42 | 档案 | 引用

GN_ghost 说：

我不知道这个署名转载时应该怎样保持署名。。。

2013年2月18日 10:52 | 档案 | 引用

Terry Zheng 说：

> 引用**Canson**的发言：
>
> 的确，或者说即使不是UEFI，Legacy的Bios的故事也有别的变种。BIOS工程师路过

能介绍一下BIOS 的工作过程么，从上电~POST 完成之间都做了什么，还有BIOS程序是怎么启动的?

2013年2月18日 16:34 | 档案 | 引用

骑驴**的桑**乔 说：

> 引用**Wht'for**的发言：
>
> >>>计算机通电后，第一件事就是读取它。

> 我更好奇，是*什么*又是*怎么*读取的。

好奇。硬件通电之后就自动去读取ROM里的BIOS程序？如何做到的？

2013年2月18日 17:07 | 档案 | 引用

yuchen 说：

看完，明白了自检和硬盘引导的道理。
谢谢阮兄～

2013年2月18日 21:50 | 档案 | 引用

fromoon 说：

一般cpu是有固化的自检的，然后北桥，没问题之后才读入bios。

2013年2月19日 13:20 | 档案 | 引用

落蓝轩主 说：

> 引用**noodles**的发言：
>
> 单个硬盘最大只能利用2T的空间
>
> 这个不是很明白

2的32次方*512Byte得出有多少Byte，然后再转化成TB。但是我不明白分区表最后四个字节为什么能够表示最多有2的32次方个扇区。

2013年2月19日 23:17 | 档案 | 引用

Jak Wings 说：

《计算机奥秘》第六版，对此也有不错的简介。

2013年2月20日 08:33 | 档案 | 引用

Jak Wings 说：

> 引用**Jak Wings**的发言：
>
> 《计算机奥秘》第六版，对此也有不错的简介。

呃，我记错了，也只是简单的说了个过程而已……不过那本书全都是彩页，页页配图，介绍了不少计算机相关的东西。

2013年2月20日 08:42 | 档案 | 引用

马小超 说：

对于第三段的第3小节，MBR 的446字节机器码就是 boot loader吧？

2013年2月20日 14:03 | 档案 | 引用

成仔不说话 说：

对于这个话题很感兴趣，根据自己了解的补充几点内存地址的东西：
1.计算机开机时，CPU默认执行0ffffh:0000h处的指令（8086是这样，386应该类似），而此内存地址应该存放的就是bios rom
2.bios执行玩post等后，将引导设备的mbr复制到内存地址07c00h处，跳转执行此处指令，这个地址应该是规定的
3.现在一般引导设备中的mbr是grub或lilo这样的引导程序，这样的程序首先将自己复制到06c00h处执行，在主分区表中搜索活动分区，将用户选择的活动分区的第一个扇区读入到07c00h处，调转到07c00h处执行
注：以上均在实模式中执行，还未进入保护模式
4.为加载内核做准备，并将控制权交给kernel，系统启动成功
注：这一步进入保护模式
当然这是现在一般的启动流程，当然如果自己写os，可以直接在引导设备的mbr中写直接加载内核的代码，将编译的代码dd入引导设备的mbr中即可。

2013年2月20日 14:57 | 档案 | 引用

熊猫家族 说：

平时都没怎么注意这么细节 在这里学习啦

2013年2月21日 10:05 | 档案 | 引用

狐说 说：

每种CPU固定的从一个地址开始运行，用于自举的程序必须固定装载在这个地方。形态可以是多种多样的

2013年2月22日 13:37 | 档案 | 引用

guisheng 说：

bootstrap不是鞋带的意思，应是靴子后面方便把靴子穿上的那个拉环

2013年2月24日 16:55 | 档案 | 引用

Javin 说：

"拽着鞋带把自己拉起来" 这个很形象。对于计算机有多了一些了解，感谢。

2013年2月25日 11:19 | 档案 | 引用

冰上游鱼 说：

从开始学计算机到现在一直想弄明白从按下计算机的启动按钮之后，每一个部件做了什么。本文讲的是大概的过程，细节恐怕很复杂，很难一一弄明白。

2013年3月 5日 01:29 | 档案 | 引用

Amy 说：

不是很详细。希望楼主继续补充。

2013年3月 6日 14:02 | 档案 | 引用

Nanthon 说：

原来Boot还有这个典故，有意思 ~

2013年3月11日 00:14 | 档案 | 引用

kiral 说：

赞，这种学习的态度值得学习。

2013年3月11日 01:58 | 档案 | 引用

Yong 说：

终于明白计算机的启动过程了，之前装Ubuntu、Win7双系统时对启动很迷茫...

2013年3月15日 14:02 | 档案 | 引用

如墨 说：

bootstrap:a loop at the back of a boot, used to pull it on.
这不就是鞋后跟上的“鞋拔子”吗？还是应该怎么叫？

补：查了一下叫“拔靴带”

2013年3月16日 04:56 | 档案 | 引用

赵胖子 说：

学习，往往这些事情都不明白，自己还不去研究，博主有心了

2013年3月17日 11:53 | 档案 | 引用

工士 说：

建议将Power-On Self-Test 翻译为 加电自检 或 开机自检

加电自检又称为开机自我检测（英文Power-on self-test，常用简称POST）

http://zh.wikipedia.org/wiki/%E5%8A%A0%E7%94%B5%E8%87%AA%E6%A3%80

2013年3月22日 13:29 | 档案 | 引用

Francise 说：

> 引用落蓝轩主的发言：
>
> 2的32次方*512Byte得出有多少Byte，然后再转化成TB。但是我不明白分区表最后四个字节为什么能够表示最多有2的32次方个扇区。

4个字节,每个字节8位二进制字符(0,1),所以总共32位,于是2的32次方

2013年4月 6日 19:20 | 档案 | 引用

Hunthon 说：

- 计算机入门选手路过，希望看到越来越多的精彩博文！

2013年5月 1日 09:21 | 档案 | 引用

林 说：

嗯 挺深奥的。
要是更清楚知道电脑的运行原理就更好；只知道它是以电为载体，就只知道0和1的信息，到底电脑是怎样识别这些数字逻辑信息的呢？如何变电流信息为电脑识别的信息。
没电脑前就只能写信 文字信息到有了电脑后的邮箱信息。
不可思议的创造，一个新鲜的东西在几十年前就出现了，而我活了二十年了还不知道那是怎么创造出来的。
求大神能解小弟疑惑啊。
万分感谢

2013年5月 5日 22:59 | 档案 | 引用

jeacon 说：

> 引用林的发言：
>
> 嗯 挺深奥的。
> 要是更清楚知道电脑的运行原理就更好；只知道它是以电为载体，就只知道0

和1的信息，到底电脑是怎样识别这些数字逻辑信息的呢？如何变电流信息为电脑识别的信息。
没电脑前就只能写信 文字信息到有了电脑后的邮箱信息。
不可思议的创造，一个新鲜的东西在几十年前就出现了，而我活了二十年了还不知道那是怎么创造出来的。
求大神能解小弟疑惑啊。
万分感谢

对于通过计算机系统的运行情况大致如下
首先源代码经过编译器变成可执行程序，可执行程序经过加载器加载到操作系统中，作为操作系统的一个进程运行，可执行程序可以理解为一条条可以被计算机CPU识别的机器码，然后CPU运行，至于CPU为什么可以运行，就需要知道数字电路中的知识，诸如加法电路，译码电路等逻辑电路知识，关于内存和CPU中的内部寄存器就需要了解数字电路中的时序电路，时序电路具备加电记忆的特点，用来实现寄存器。
数字电路研究的是0101这种高低电平之间转换和输入输出的规律。
至于数字电路的基础就要是模拟电路了，二极管三极管等等。大量的三极管构成了一个基本的数字电路单元，例如能够存储一位二进制数的寄存器。
而模拟电路研究的是大量电子在加压后的宏观运动规律，这种宏观体现到数字电路上就是01的高低电平。
那么单个电子的运动规律是如何的呢？这就需要电子量子学的知识作为支撑了，但是现在科技能够研究到的最深的领域，也就是量子了（电子也属于量子），至于我们以前所学习过的单个量子的运行规律不见得是对的，但是这并不影响到我们研究大量量子的运行规律，单个量子不好研究，但是大量量子就会呈现出来一定的规律性。

当你研究到量子，你也就研究到了当今科学最前沿的东西了，也就没有办法更深了。。。

不知我说的是否容易理解。

2013年5月 8日 13:53 | 档案 | 引用

techon 说：

这是计算机的常规过程，关于启动这块现在正处于老式和BIOS-MBR和UEFI-GPT的

过度阶段

wiki上 UEFI的介绍：
https://zh.wikipedia.org/wiki/UEFI

前一阵看到的一篇文章：新一代UEFI BIOS科普和探索系列
http://sphrbeu2012.blog.163.com/blog/static/209228074201331472248600/

2013年5月12日 01:12 | 档案 | 引用

冰与火 说：

从您这儿了解到，为什么只能有4个主分区了。我之前想对硬盘分更多分区的时候（Windows 7），找到的解决办法是change the basic disk to dynamic disk on Disk 0.
当时成功的实现了扩展分区，很高兴。但并没有深究这个convertion对电脑又什么影响。
我觉得应该像您学习，追根溯源。
但是，我从网上也只是了解到Basic和Dynamic之间定义的区别，并没看到转换之间的区别。Windows社区里倒是有老外问起这个问题，似乎也得到了非常满意的答案，只是我没读懂那段英文的意思。http://social.technet.microsoft.com/Forums/en-US/winservergen/thread/cc90e135-4aa3-4ba0-8b22-057566dba5ec
有对此清楚的朋友，还望赐教！

2013年6月 9日 10:37 | 档案 | 引用

niurg 说：

写的非常好，可惜没有从头至尾学过，很多环节都很模糊。

2013年6月17日 13:46 | 档案 | 引用

张启杰 说：

有一个文档“From Power UP to Bash Prompt”很好地描述了计算机启动的过程，可以到网上搜来看看。

2013年7月 4日 16:00 | 档案 | 引用

shincepu 说：

高质量博文 + 高质量回复。谢谢各位了

2013年8月17日 23:36 | 档案 | 引用

peng 说：

> 引用马小超的发言：
>
> 对于第三段的第3小节，MBR 的446字节机器码就是 boot loader吧？

我记得也是这样 ，boot loader可以执行标准的动作来启动 ，但也可以干些其他事用其他方式启动， 就像grub那样，grub就是bootloader。

2013年9月23日 20:23 | 档案 | 引用

李健 说：

老师：可以把你写这些文章的时候顺便把你的那些参考资料也记载一下吧。这样在看你的文章的时候也可以多学点，真心觉得写得不错。。

2013年9月30日 00:45 | 档案 | 引用

篱下殇 说：

"所以单个硬盘可利用的空间最大也不超过2TB"这个没理解，应该是硬盘分区最大为2T吧

2013年10月 6日 18:26 | 档案 | 引用

Melo 说：

本篇博文主要还是讲述的是intel的x86架构的启动原理，实际上arm启动原理与上述过程并不相同。

2013年10月19日 12:09 | 档案 | 引用

Melo 说：

x86为了兼容以前的8086CPU，所以开机时进入实模式，实模式下是16位寻址方式，此时0ffffh:0000h的物理地址指向了32位寻址范围的4G内存的最后64K的16字节处，正是BIOS的ROM代码，BIOS的存在意义主要是建立一个中断向量表，然后用这些中断去检测和读取其他的设备，比如内存，硬盘等，最后会被操作系统的代码重新覆盖这些向量。

2013年10月19日 12:17 | 档案 | 引用

孟帅 说：

这是Linux启动后的第一个进程，pid进程编号为1，其他进程都是它的后代。
对于这句话 我不太理解 怎么叫“其他进程都是它的后代”？是不是说，电脑开着的时候，这个进程一直在运行，一直占用着内存？后代用来比喻什么？进程之间的关系哪有后代之说？那不就是线程吗？

2013年11月30日 20:08 | 档案 | 引用

dliyc 说：

@jeacon：

解释到数字电路就可以了，诸如触发器，加法器之类的。量子嘛，就扯的有点远了。

2013年12月22日 21:28 | 档案 | 引用

鬼鬼 说：

扩展分区可以包含无数个逻辑分区 这个有限制的吧，不是无限的

2014年1月 6日 15:38 | 档案 | 引用

Song 说：

博主的文章 + “成仔不说话”的回复，对我的帮助最大。
博主让我知道了boot的真正含义，以前确实困惑过加载时“蛋和鸡”的问题。
接着对于ROM,BIOS,MBR512,4硬盘,2TB,都能理解。
就是困惑谁执行拷贝，搬运操作系统到内存，怎么交接控制权有点模糊。
“成仔不说话”的补充回答了我这个疑惑，包括：

“CPU默认执行0ffffh:0000h处的指令，rom-bios”
“将引导设备的mbr复制到内存地址07c00h处，跳转执行此处指令”
“grub或lilo这样的引导程序，先将自己复制到06c00h处执行，搜索活动分区读入到07c00h处，调转到07c00h处执行，控制权交给kernel”
kernel开始启动自己第一个进程init...
总结：
1）拷贝都是由前一个运行的程序执行。
2）控制权的交接是拷贝后执行跳转（应该是通过CPU的PC）。

2014年1月11日 12:59 | 档案 | 引用

路易十六 说：

怎么和鸟哥说的相左。假如，我双系统都安装在主分区，咋办？？

2014年3月10日 20:09 | 档案 | 引用

the 说：

版主写的计算机启动是针对windows系统吗？
另外，文章中提到的分区表、主分区及逻辑分区这些概念也是针对windows系统吗？

2014年3月19日 18:29 | 档案 | 引用

北平Tel 说：

相当不错啊；不过，你不可以用 Linux 系统举例。毕竟，我们大部分人都是计算机菜鸟，用了一辈子 Windows :)

2014年5月30日 01:41 | 档案 | 引用

luciuz 说：

> 引用北平℡的发言：
>
> 相当不错啊；不过，你不可以用 Linux 系统举例。毕竟，我们大部分人都是计算机菜鸟，用了一辈子 Windows :)

说反了吧您？ 看到windows就觉得务必讨厌。码农应该和我有同感吧

2014年7月10日 19:49 | 档案 | 引用

fyzjhh 说：

有个疑惑，当选择了一个启动设备之后，如何决定是 分区启动（3.1 情况A：卷引导记录） 还是 启动管理器（3.3 情况C：启动管理器） 启动呢？

2014年9月16日 11:06 | 档案 | 引用

fyzjhh 说：

相当不错啊；不过，你不可以用 Linux 系统举例。毕竟，我们大部分人都是计算机菜鸟，用了一辈子.

2014年10月11日 13:46 | 档案 | 引用

loszer 说：

非常感谢，很清晰呢，受益匪浅

2014年12月 8日 20:25 | 档案 | 引用

水中月镜中花 说：

计算机先读取扩展分区的第一个扇区，叫做"扩展引导记录"（Extended boot record，缩写为EBR）。它里面也包含一张64字节的分区表，但是最多只有两项（也就是两个逻辑分区）。
计算机接着读取第二个逻辑分区的第一个扇区，再从里面的分区表中找到第三个逻辑分区的位置，以此类推，直到某个逻辑分区的分区表只包含它自身为止（即只有一个分区项）。因此，扩展分区可以包含无数个逻辑分区。
但是，似乎很少通过这种方式启动操作系统。如果操作系统确实安装在扩展分区，一般采用下一种方式启动。
这段话没有看懂
-----前面一句话是说扩展分区的第一个扇区最多有两个逻辑分区还是说整个扩展分区最多只能有两个逻辑分区呢？

2014年12月12日 22:08 | 档案 | 引用

Zhiqiang He 说：

>以Linux系统为例，先载入/boot目录下面的kernel。内核加载成功后，第一个运行的程序是/sbin/init。它根据配置文件（Debian系统是/etc/initab）产生init进程。这是Linux启动后的第一个进程，pid进程编号为1，其他进程都是它的后代。

刚刚看了下，Debian init配置文件是 /etc/inittab ,文中应该有个小错误

2015年1月23日 19:22 | 档案 | 引用

newbie 说：

很难让我了解了下启动的情节，厉害！！不过现在UEFI流行起来了，希望能看到有关UEFI的启动方面的文章！

2015年2月16日 20:23 | 档案 | 引用

我要发表看法

您的留言 （HTML标签部分可用）

您的大名：

«-必填

电子邮件：

«-必填，不公开

个人网址：

«-我信任你，不会填写广告链

接

记住个人信息？

发表 «- 点击按钮