

## 07 | 从BIOS到bootloader：创业伊始，有活儿老板自己上

2019-04-10 刘超

趣谈Linux操作系统

[进入课程 >](#)



讲述：刘超

时长 12:55 大小 11.84M



有了开放的营商环境，咱们外包公司的创业之旅就要开始了。

上一节我们说，x86 作为一个开放的营商环境，有两种模式，一种模式是实模式，只能寻址 1M，每个段最多 64K。这个太小了，相当于咱们创业的个体户模式。有了项目只能老板自己上，本小利微，万事开头难。另一种是保护模式，对于 32 位系统，能够寻址 4G。这就是大买卖了，老板要雇佣很多人接项目。

几乎所有成功的公司，都是从个体户模式发展壮大的，因此，这一节咱们就从系统刚刚启动的个体户模式开始说起。

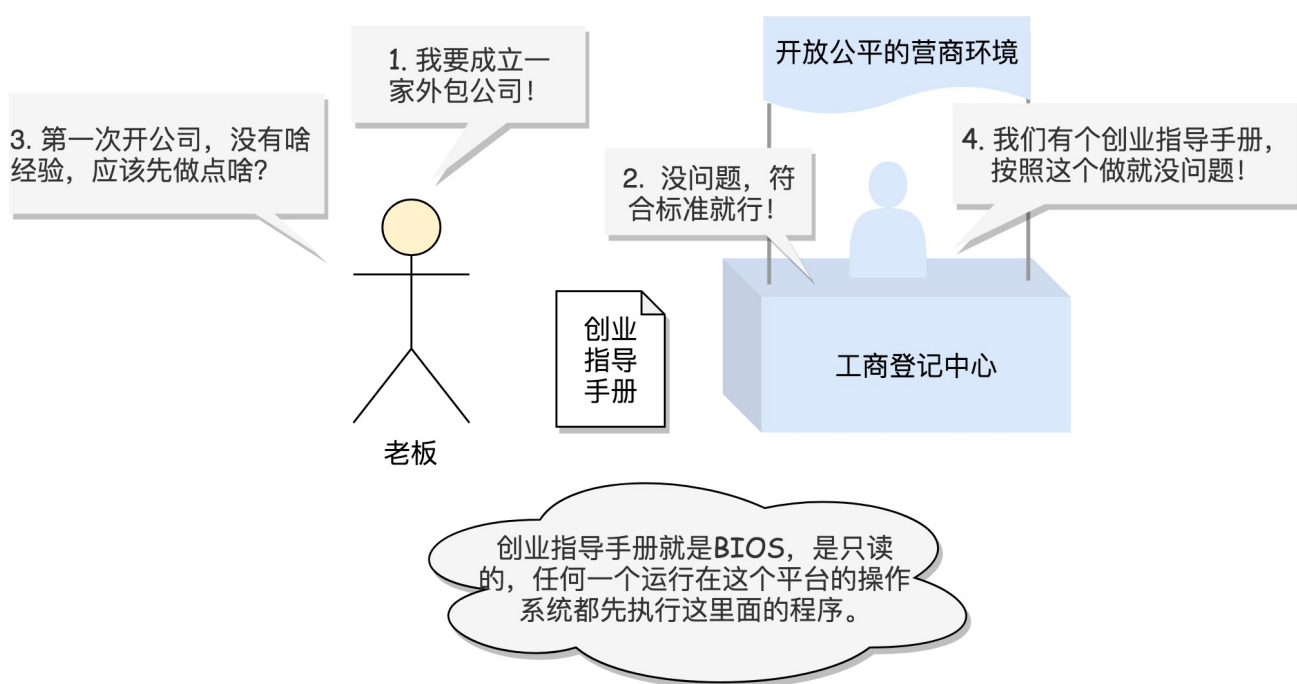
### BIOS 时期

当你轻轻按下计算机的启动按钮时，你的主板就加上电了。

按照我们之前说的，这时候你的 CPU 应该开始执行指令了。你作为老板，同时也作为员工，要开始干活了。可是你发现，这个时候还没有项目执行计划书，所以你没啥可干的。

也就是说，这个时候没有操作系统，内存也是空的，一穷二白。CPU 该怎么办呢？

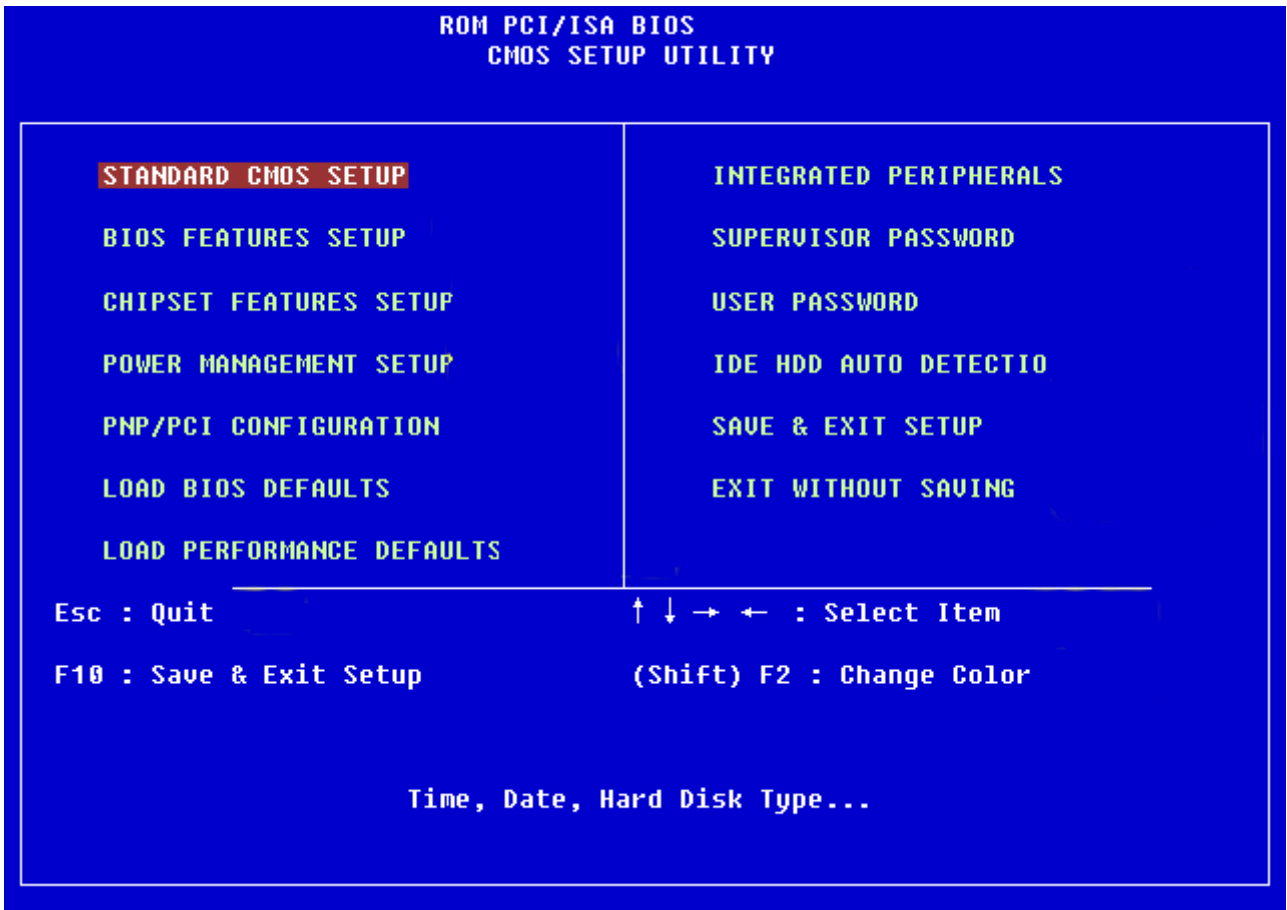
你作为这个创业公司的老板，由于原来没开过公司，对于公司的运营当然是一脸懵的。但是我们有一个良好的营商环境，其中的创业指导中心早就考虑到这种情况了。于是，创业指导中心就给了你一套创业公司启动指导手册。你只要按着指导手册来干就行了。



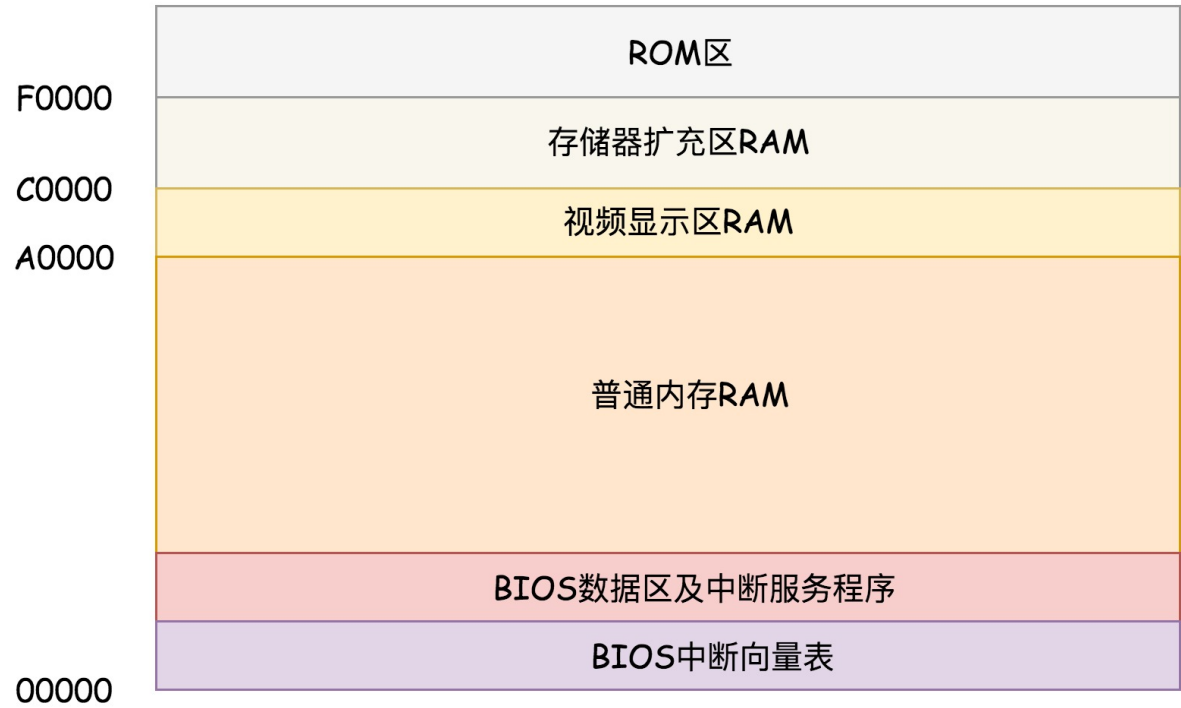
计算机系统也早有计划。在主板上，有一个东西叫**ROM**（Read Only Memory，只读存储器）。这和咱们平常说的内存**RAM**（Random Access Memory，随机存取存储器）不同。

咱们平时买的内存条是可读可写的，这样才能保存计算结果。而 ROM 是只读的，上面早就固化了一些初始化的程序，也就是**BIOS**（Basic Input and Output System，基本输入输出系统）。

如果你自己安装过操作系统，刚启动的时候，按某个组合键，显示器会弹出一个蓝色的界面。能够调整启动顺序的系统，就是我说的 BIOS，然后我们就可以先执行它。



创业初期，你的办公室肯定很小。假如现在你有 1M 的内存地址空间。这个空间非常有限，你需要好好利用才行。



在 x86 系统中，将 1M 空间最上面的 0xF0000 到 0xFFFFF 这 64K 映射给 ROM，也就是说，到这部分地址访问的时候，会访问 ROM。

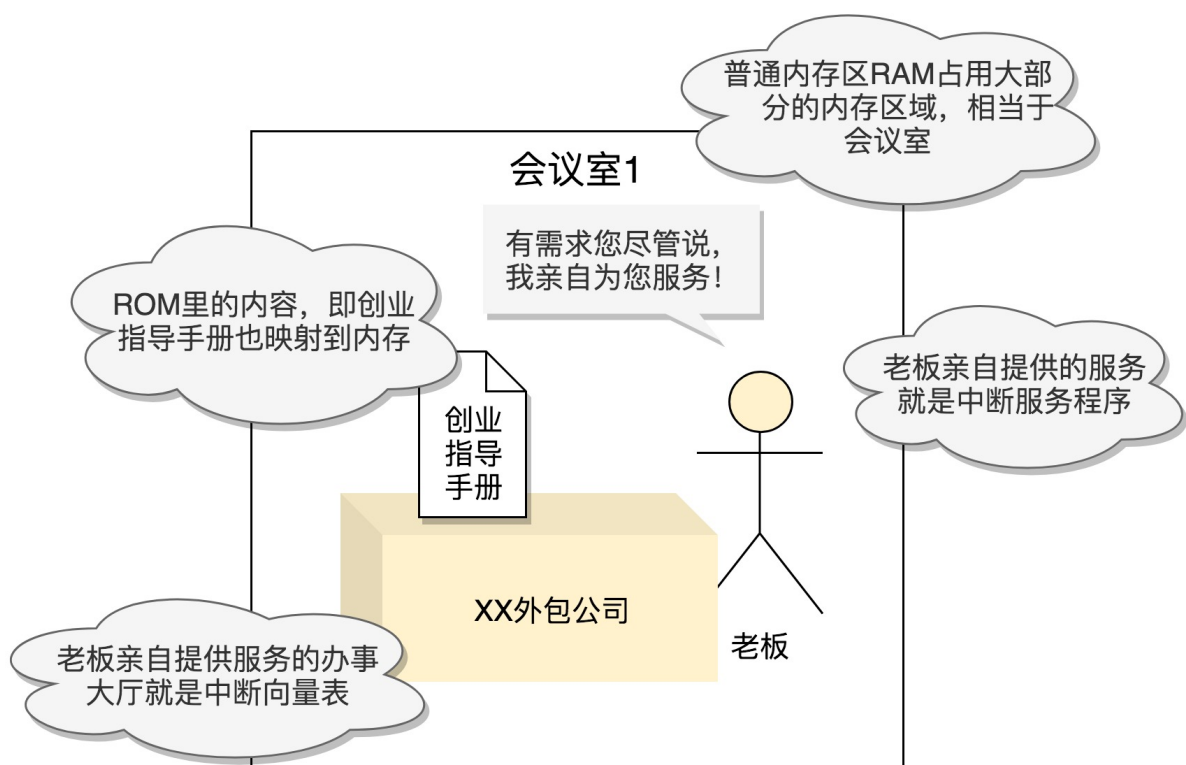
当电脑刚加电的时候，会做一些重置的工作，将 CS 设置为 0xFFFF，将 IP 设置为 0x0000，所以第一条指令就会指向 0xFFFF0，正是在 ROM 的范围内。在这里，有一个 JMP 命令会跳到 ROM 中做初始化工作的代码，于是，BIOS 开始进行初始化的工作。

创业指导手册第一条，BIOS 要检查一下系统的硬件是不是都好着呢。

创业指导手册第二条，要有个办事大厅，只不过自己就是办事员。这个时期你能提供的服务很简单，但也会有零星的客户来提要求。

这个时候，要建立一个中断向量表和中断服务程序，因为现在你还要用键盘和鼠标，这些都要通过中断进行的。

这个时期也要给客户输出一些结果，因为需要你自己来，所以你还要充当客户对接人。你做了什么工作，做到了什么程度，都要主动显示给客户，也就是在内存空间映射显存的空间，在显示器上显示一些字符。



最后，政府领进门，创业靠个人。接下来就是你发挥聪明才智的时候了。

## bootloader 时期


政府给的创业指导手册只能保证你把公司成立起来，但是公司如何做大做强，需要你自己有一套经营方法。你可以试着从档案库里面翻翻，看哪里能够找到《企业经营宝典》。通过这个宝典，可以帮你建立一套完整的档案库管理体系，使得任何项目的档案查询都十分方便。

现在，什么线索都没有的 BIOS，做完自己的事情，只能从档案库门卫开始，慢慢打听操作系统的下落。

操作系统在哪儿呢？一般都会在安装在硬盘上，在 BIOS 的界面上。你会看到一个启动盘的选项。启动盘有什么特点呢？它一般在第一个扇区，占 512 字节，而且以 0xAA55 结束。这是一个约定，当满足这个条件的时候，就说明这是一个启动盘，在 512 字节以内会启动相关的代码。

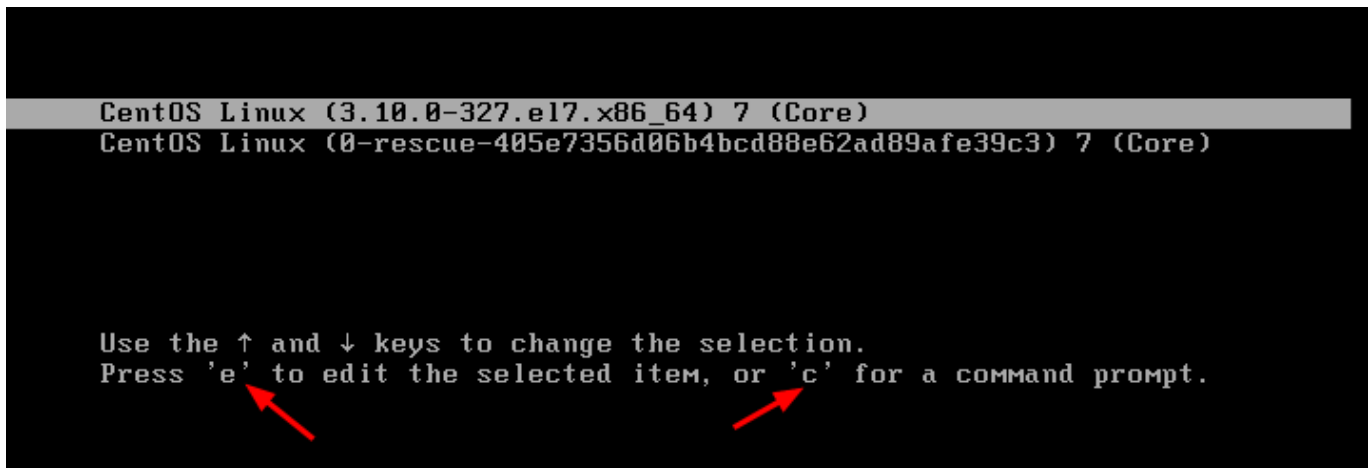
这些代码是谁放在这里的呢？在 Linux 里面有一个工具，叫**Grub2**，全称 Grand Unified Bootloader Version 2。顾名思义，就是搞系统启动的。

你可以通过 `grub2-mkconfig -o /boot/grub2/grub.cfg` 来配置系统启动的选项。你可以看到里面有类似这样的配置。

 复制代码

```
1 menuentry 'CentOS Linux (3.10.0-862.el7.x86_64) 7 (Core)' --class centos --class gnu-li
2     load_video
3     set gfxpayload=keep
4     insmod gzio
5     insmod part_msdos
6     insmod ext2
7     set root='hd0,msdos1'
8     if [ x$feature_platform_search_hint = xy ]; then
9         search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1' b1aceb95-6b9e-464a-a589-bed66220ebee
10    else
11        search --no-floppy --fs-uuid --set=root b1aceb95-6b9e-464a-a589-bed66220ebee
12    fi
13    linux16 /boot/vmlinuz-3.10.0-862.el7.x86_64 root=UUID=b1aceb95-6b9e-464a-a589-bed66220ebee
14    initrd16 /boot/initramfs-3.10.0-862.el7.x86_64.img
15 }
```

这里面的选项会在系统启动的时候，成为一个列表，让你选择从哪个系统启动。最终显示出来的结果就是下面这张图。至于上面选项的具体意思，我们后面再说。



使用 `grub2-install /dev/sda`，可以将启动程序安装到相应的位置。

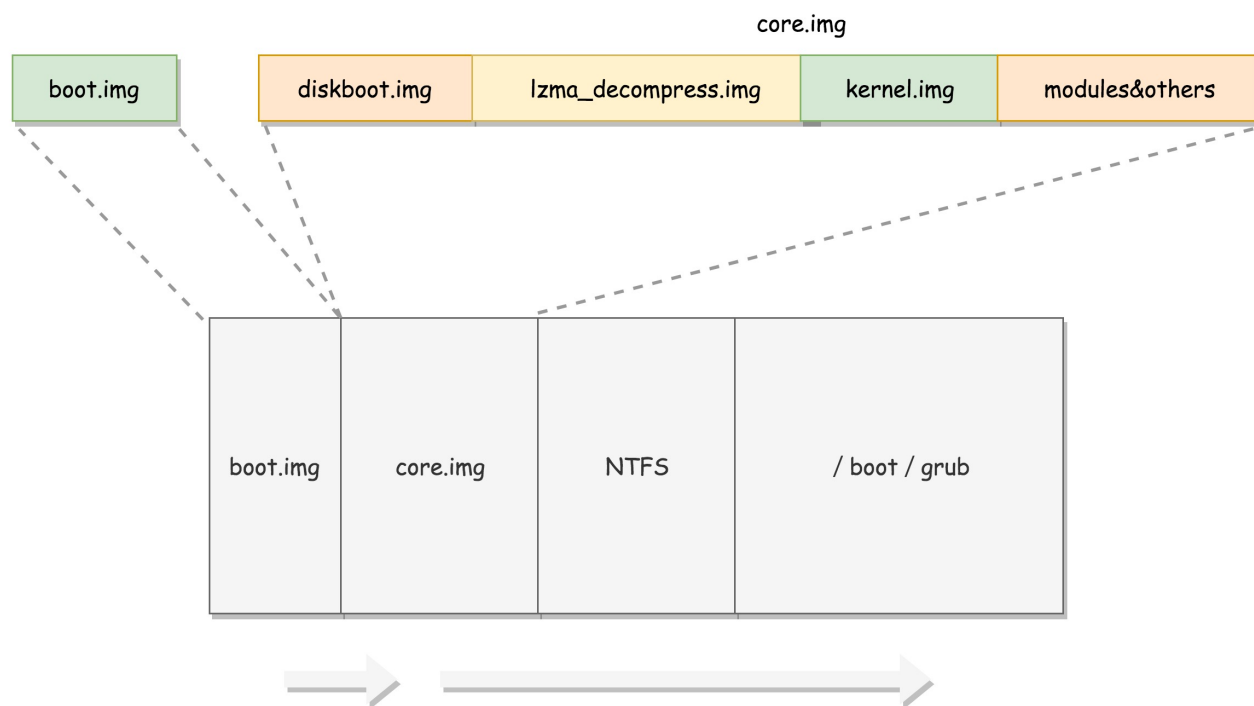
grub2 第一个要安装的就是 `boot.img`。它由 `boot.S` 编译而成，一共 512 字节，正式安装到启动盘的第一个扇区。这个扇区通常称为 **MBR** (Master Boot Record, 主引导记录 / 扇区)。

BIOS 完成任务后，会将 `boot.img` 从硬盘加载到内存中的 `0x7c00` 来运行。

由于 512 个字节实在有限，`boot.img` 做不了太多的事情。它能做的最重要的一个事情就是加载 grub2 的另一个镜像 `core.img`。

引导扇区就是你找到的门卫，虽然他看着档案库的大门，但是知道的事情很少。他不知道你的宝典在哪里，但是，他知道应该问谁。门卫说，档案库入口处有个管理处，然后把你领到门口。

`core.img` 就是管理处，它们知道的和能做的事情就多了一些。`core.img` 由 `lzma_decompress.img`、`diskboot.img`、`kernel.img` 和一系列的模块组成，功能比较丰富，能做很多事情。



boot.img 先加载的是 core.img 的第一个扇区。如果从硬盘启动的话，这个扇区里面是 diskboot.img，对应的代码是 diskboot.S。

boot.img 将控制权交给 diskboot.img 后，diskboot.img 的任务就是将 core.img 的其他部分加载进来，先是解压缩程序 lzma\_decompress.img，再往下是 kernel.img，最后是各个模块 module 对应的映像。这里需要注意，它不是 Linux 的内核，而是 grub 的内核。

lzma\_decompress.img 对应的代码是 startup\_raw.S，本来 kernel.img 是压缩过的，现在执行的时候，需要解压缩。

在这之前，我们所有遇到过的程序都非常非常小，完全可以在实模式下运行，但是随着我们加载的东西越来越大，实模式这 1M 的地址空间实在放不下了，所以在真正的解压缩之前，lzma\_decompress.img 做了一个重要的决定，就是调用 real\_to\_prot，切换到保护模式，这样就能在更大的寻址空间里面，加载更多的东西。

## 从实模式切换到保护模式

好了，管理处听说你要找宝典，知道你将来是要做老板的人。既然是老板，早晚都要雇人干活的。这不是个体户小打小闹，所以，你需要切换到老板角色，进入保护模式了，把哪些是

你的权限，哪些是你可以授权给别人的，都分的清清楚楚。

切换到保护模式要干很多工作，大部分工作都与内存的访问方式有关。

第一项是**启用分段**，就是在内存里面建立段描述符表，将寄存器里面的段寄存器变成段选择子，指向某个段描述符，这样就能实现不同进程的切换了。第二项是**启动分页**。能够管理的内存变大了，就需要将内存分成相等大小的块，这些我们放到内存那一节详细再讲。

切换到了老板角色，也是为了招聘很多人，同时接多个项目，这时候就需要划清界限，懂得集权与授权。

当了老板，眼界要宽多了，同理保护模式需要做一项工作，那就是打开 Gate A20，也就是第 21 根地址线的控制线。在实模式 8086 下面，一共就 20 个地址线，可访问 1M 的地址空间。如果超过了这个限度怎么办呢？当然是绕回来了。在保护模式下，第 21 根要起作用了，于是我们就需要打开 Gate A20。

切换保护模式的函数 `DATA32 call real_to_prot` 会打开 Gate A20，也就是第 21 根地址线的控制线。

现在好了，有的是空间了。接下来我们要对压缩过的 `kernel.img` 进行解压缩，然后跳转到 `kernel.img` 开始运行。

切换到了老板角色，你可以正大光明地进入档案馆，寻找你的那本宝典。

`kernel.img` 对应的代码是 `startup.S` 以及一堆 `c` 文件，在 `startup.S` 中会调用 `grub_main`，这是 `grub` `kernel` 的主函数。

在这个函数里面，`grub_load_config()` 开始解析，我们上面写的那个 `grub.conf` 文件里的配置信息。

如果是正常启动，`grub_main` 最后会调用 `grub_command_execute ("normal" , 0, 0)`，最终会调用 `grub_normal_execute()` 函数。在这个函数里面，`grub_show_menu()` 会显示出让你选择的那个操作系统的列表。



同理，作为老板，你发现这类的宝典不止一本，经营企业的方式也有很多种，到底是人性化的，还是强纪律的，这个时候你要做一个选择。

一旦，你选定了某个宝典，启动某个操作系统，就要开始调用 `grub_menu_execute_entry()`，开始解析并执行你选择的那一项。接下来你的经营企业之路就此打开了。

例如里面的 `linux16` 命令，表示装载指定的内核文件，并传递内核启动参数。于是 `grub_cmd_linux()` 函数会被调用，它会首先读取 Linux 内核镜像头部的一些数据结构，放到内存中的数据结构来，进行检查。如果检查通过，则会读取整个 Linux 内核镜像到内存。

如果配置文件里面还有 `initrd` 命令，用于为即将启动的内核传递 `init ramdisk` 路径。于是 `grub_cmd_initrd()` 函数会被调用，将 `initramfs` 加载到内存中来。

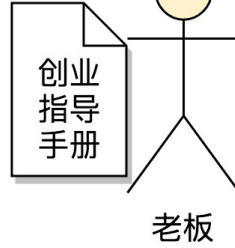
当这些事情做完之后，`grub_command_execute ("boot", 0, 0)` 才开始真正地启动内核。

## 总结时刻

启动的过程比较复杂，我这里画一个图，让你比较形象地理解这个过程。你可以根据我讲的，自己来梳理一遍这个过程，做到不管是从流程还是细节上，都能心中有数。

BIOS

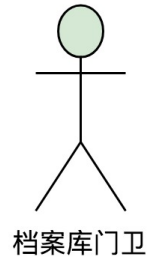
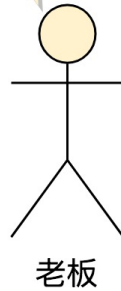
我得找到《企业经营宝典》！



引导扇区  
boot.img

档案库里有《企业经营宝典》吗？

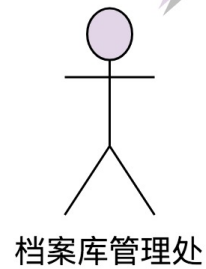
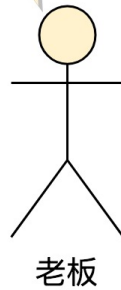
我是门卫，我不清楚，  
你去问管理处吧！



diskboot.img

档案库里有《企业经营宝典》吗？

你要做老板呀？



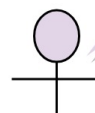
lzma\_decompress.img

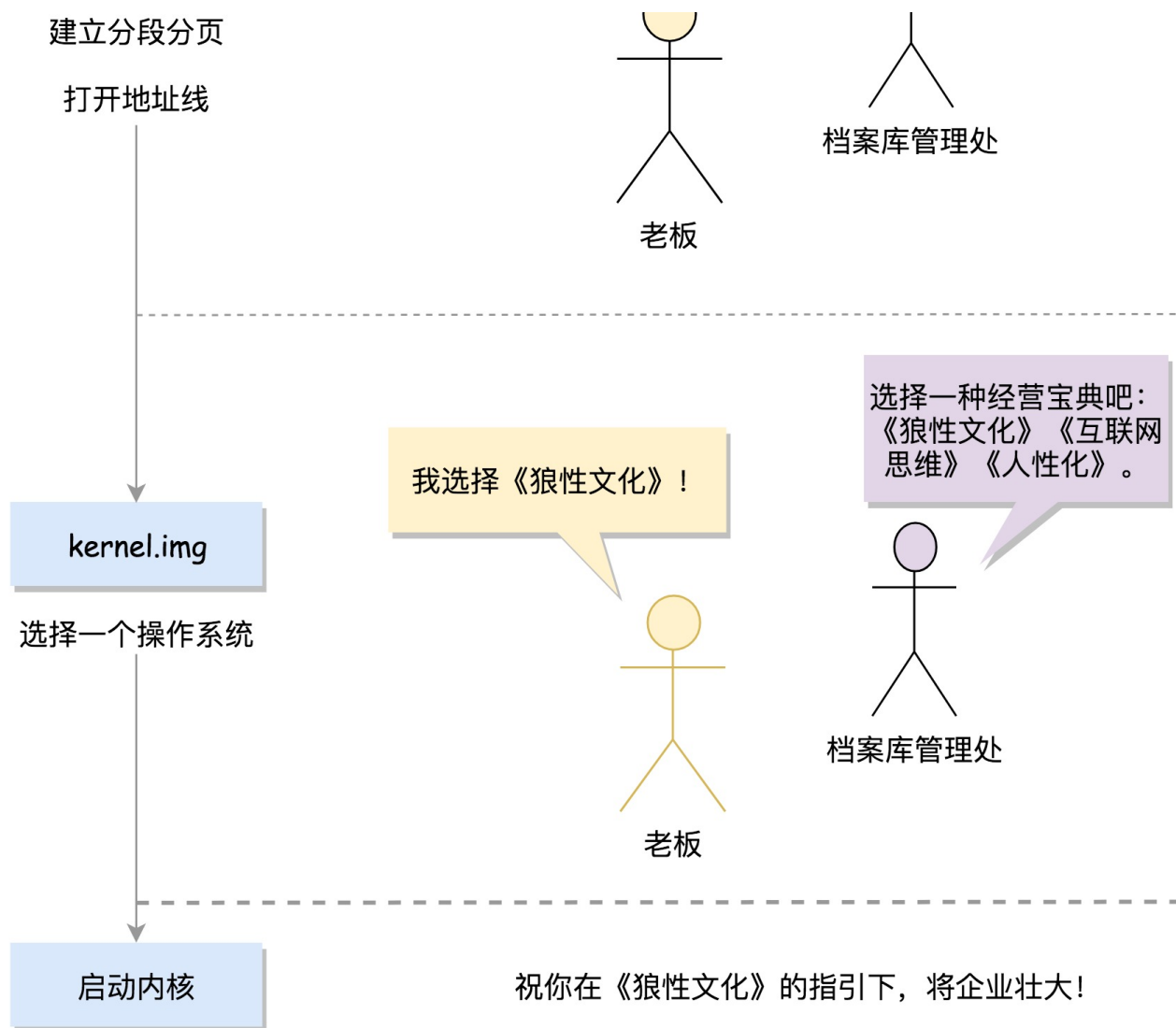
实模式到保护模式

好的！

我可以领你去找宝典。

注意以后你就是老板了，眼界  
宽阔些，注意集权和授权！





## 课堂练习

grub2 是一个非常牛的 Linux 启动管理器，请你研究一下 grub2 的命令和配置，并试试通过它启动 Ubuntu 和 CentOS 两个操作系统。

欢迎留言和我分享你的疑惑和见解，也欢迎你收藏本节内容，反复研读。你也可以把今天的内容分享给你的朋友，和他一起学习、进步。

# 趣谈 Linux 操作系统

像故事一样的操作系统入门课

刘超

网易杭州研究院

云计算技术部首席架构师



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 06 | x86架构：有了开放的架构，才能打造开放的营商环境

下一篇 08 | 内核初始化：生意做大了就得成立公司

## 精选留言 (62)

写留言



why

2019-04-10

72

- 实模式只有 1MB 内存寻址空间(X86)
- 加电, 重置 CS 为 0xFFFF, IP 为 0x0000, 对应 BIOS 程序
- 0xF0000-0xFFFFF 映射到 BIOS 程序(存储在ROM中), BIOS 做以下三件事:
  - 检查硬件
  - 提供基本输入(中断)输出(显存映射)服务...

展开 ▾



我爱北京天...

2019-04-10

25

看来从这篇开始我要看三遍四遍五遍的节奏了

展开 ▾



Luke

2019-04-11

👍 20

这部分的实验，大家可以去github看我的工程哈，icecoobe/oslab，已经进入保护模式了，还有很远的路，一起加油！

作者回复: 牛



Luke

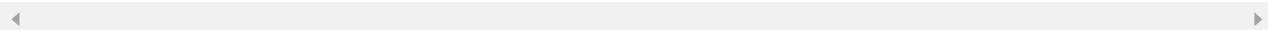
2019-04-11

👍 14

看到很多人留言需要资料，我来推荐一本新书《一个64位操作系统的设计与实现》，如果你有汇编基础，很感兴趣底层的细节，可以看李忠的那本《从实模式到保护模式》

展开 ▾

作者回复: 赞，看来我得收集一下书名，统一推荐给大家



Li Shundu...

2019-04-10

👍 13

老板选择了《狼性文化》😁😁

展开 ▾



赵又新

2019-04-11

👍 6

之前课上说的，如果没有理解错的话：

32位，分为16位寻址空间和16位偏移量。但通过左移4位的方式，将寻址空间扩充为20位。所以，0xFFFF的位置实际指的是0xFFFF0。

展开 ▾

作者回复: 是的





影影影

2019-04-10

👍 6

补充阅读

<https://opensource.com/article/17/2/linux-boot-and-startup>

<https://opensource.com/article/17/3/introduction-grub2-configuration-linux>

---



wahaha

2019-04-12

👍 4

grub2 是一个非常牛的 Linux 启动管理器

这句应该去掉Linux, 因为GRUB2也能启动其它操作系统

---



随风

2019-04-10

👍 4

当电脑刚加电的时候, 会做一些重置的工作, 将 CS 设置为 0xFFFF, 将 IP 设置为 0x0000, 所以第一条指令就会指向 0xFFFF0。这个所以怎么得到的结果? 为什么上面都是五位 0xFFFFF, cs/ip都是四位0xFFFF? 小白越看越不明白了。

展开 ▾

---



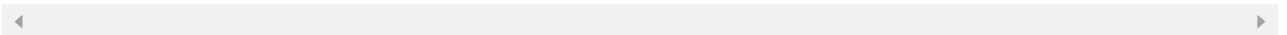
天使也有爱

2019-04-10

👍 4

老师 我现在看这些内容有点晕 太细了 我是要用那本书做配套看 还是直接用内核源码结合着看呢

作者回复: 我推荐了书籍, 对着源码看挺好的



Socrakit

2019-04-16

👍 3

查了一些资料, 关于 Gate A20 我的理解是:

- 8086 地址线20根 -> 可用内存 0 ~ FFFFF

寄存器却是16位, 寻址模式为 segment(16位):offset(16位), 最大范围变成 0FFFF0(左移了4位) + 0FFFF = 10FFEF...

展开 ▾

---



leon

2019-04-14

👍 3

32位处理器不是有32根地址线嘛？为啥只打开第21根地址线的控制线？这里可以再稍微解释一下吗？控制线是另外一种线嘛？

---



天王

2019-04-11

👍 3

总结:ROM只读存储器，ROM固化了一些程序就是BIOS，用来初始化系统，一开始的内存空间比较小，只有1M，最上面的64k映射为BIOS，指针指向这64k，开始进行初始化，有2个事情，一个是检查硬件环境，另一个是建立中断程序和中断向量表，同时把结果显示在显示器上，BIOS只是做初始化工作，真正安装系统了，首先要找系统，grub2是搞系统启动的，他把系统代码放在硬盘上，一般在第一个扇区，以0xAA55结束，512个字节，满...  
展开 ∨

---



徐庆新

2019-04-11

👍 3

RAM是Random Access Memory，不是Read Access Memory

展开 ∨

作者回复: 我读的是read?



Zach\_

2019-04-10

👍 3

- 1.BIOS时期约定了启动扇区的位置与大小
- 2.BootLoader时期主要是通过grub2来启动系统
- 3.启动过程中有实模式到保护模式的切换。

展开 ∨

---



流殇忘情

2019-04-11

👍 2

既然BIOS是只读的，那升级BIOS固件是怎么做到的呢？

展开 ∨

作者回复: 写入方式不一样, 咱们不是说升级bios是烧bios么



TeFuir

2019-04-10

👍 2

当电脑刚加电的时候, 会做一些重置的工作, 将 CS 设置为 0xFFFF, 将 IP 设置为 0x0000, 所以第一条指令就会指向 0xFFFF0, 正是在 ROM 的范围内。为什么第一条指令会指向0xFFFF0呢

展开 ▾

作者回复: 左移四位



garlic

2019-05-22

👍 1

grub2-mkconfig 可以通过30\_os-prober 发现安装的其他系统, 也可以通过修改 40\_custom文件配置chainloader 实现 多操作系统引导, 笔记链接.

<https://garlicspace.com/2019/05/17/grub2-配置centos7%ef%bc%8cubuntu/>



yan华建

2019-05-12

👍 1

个人小结

ROM(只读存储器) read only memory

RAM(random access memory)随机存取存储器

BIOS(Basic Input and Output System)基本输入输出系统

MBR(Master Boot Record)主引导分区/记录...

展开 ▾



小松松

2019-04-23

👍 1

感觉要看懂这个专栏, 我先要学习好另一个专栏 《深入浅出计算机组成原理》

展开 ▾



作者回复: 赞，其实不用全学完，学一部分就可以回来

