<<30天自制操作系统>>

图书基本信息

书名：<<30天自制操作系统>>

13位ISBN编号：9787115287960

10位ISBN编号：7115287961

出版时间：2012-8

出版时间：人民邮电出版社

作者：[日] 川合秀实

页数：710

字数：1063000

译者：周自恒,李黎明,曾祥江,张文旭

前言

“好想编写一个操作系统呀！”笔者的朋友曾说这是所有程序员都曾经怀揣的一个梦想。说“所有的程序员”可能有点夸张了，不过作为程序员的梦想，它至少也应该能排进前十名吧。

也许很多人觉得编写操作系统是个天方夜谭，这一定是操作系统业界的一个阴谋（笑）。他们故意让大家相信编写操作系统是一件非常困难的事情，这样就可以高价兜售自己开发的操作系统，而且操作系统的作者还会被顶礼膜拜。

那么实际情况又怎么样呢？

和别的程序相比，其实编写操作系统并没有那么难，至少笔者的感觉是这样。

在各位读者之中，也许有人曾经挑战过操作系统的编写，但因为太难而放弃了。拥有这样经历的人也许不会认同笔者的观点。

其实你错了，你的失败并不是因为编写操作系统太难，而是因为没有人告诉你那其实是一件很简单的事而已。

不仅是编写操作系统，任何事都是一样的。

如果讲解的人认为它很难，那就不可能把它讲述得通俗易懂，即便是同样的内容，也会讲得无比复杂。这样的讲解，肯定是很难懂的。

那么，你想不想和笔者一起再挑战一次呢？

如果你曾经梦想过编写自己的操作系统，一定会觉得乐在其中的。

可能有人会说，这本书足足有700多页，怎么会“有趣”和“简单”呢？唔，这么一说笔者也觉得挺心虚的，不过其实也只是长了那么一点点啦。平均下来的话，每天只有大约23页的内容，你看，也没有那么长吧？

这本书的文风非常轻松，也许你不知不觉中就会读得很快。但是这样的话可能印象不会很深，最好还是能静下心来慢慢地读。

书中所展示的程序代码和文字的说明同样重要，因此也希望大家仔细阅读。只要注意这些，理解本书的内容就应该没有问题了。

在本书中，我们使用C语言和汇编语言来编写操作系统，不过不必担心，你可以在阅读本书的同时来逐步学习关于这些编程语言的知识。本书在这方面写得非常仔细，如果能有人通过本书终于把C语言中的指针给搞懂了，那笔者的目的也就达到了。

即便是从这样的水平开始，30天后你也能够编写出一个很棒的操作系统，请大家拭目以待吧！

<<30天自制操作系统>>

内容概要

自己编写一个操作系统，是许多程序员的梦想。
也许有人曾经挑战过，但因为太难而放弃了。
其实你错了，你的失败并不是因为编写操作系　统太难，而是因为没有人告诉你那其实是一件很简单的事。
那么，你想不想再挑战一次呢？

这是一本兼具趣味性、实用性与学习性的书籍。
作者从计算机的构造、汇编语言、C语言开始解说，让你在实践中掌握算法。
在这本书的指导下，从零编写所有代码，30天后就可以制作出一个具有窗口系统的32位多任务操作系统。

本书以课题为主导，边做边玩，抛开晦涩难懂的语言，行文风格十分随性，还充满了各种欢乐的吐槽，适合操作系统爱好者和程序设计人员阅读。

作者简介

川合秀实（Hidemi Kawai）
生于1975年，是一位以“轻量化”编程思想见长的“非主流”开发者。
2000年因自行开发的OSASK项目而名声大噪。
OSASK是一个开源的32位微型操作系统，它并非以Linux等内核为基础，而是完全从零开始开发，在一张软盘的容量下实现了GUI、多任务、多语言等高级特性，启动时间只需1秒。
本书的内容可以看成是作者以OSASK为蓝本，教会读者从零开始开发一个操作系统，同时可以让初学者在编写操作系统的过程中，了解操作系统背后更多的知识。

<<30天自制操作系统>>

书籍目录

章节摘录

版权页： 插图： 现在我们要进行内存管理了。
首先必须要做的事情，是搞清楚内存究竟有多大，范围是到哪里。
如果连这一点都搞不清楚的话，内存管理就无从谈起。
在最初启动时，BIOS肯定要检查内存容量，所以只要我们问一问BIOS，就能知道内存容量有多大。
但问题是，如果那样做的话，一方面asmhead.nas会变长，另一方面，BIOS版本不同，BIOS函数的调用方法也不相同，麻烦事太多了。
所以，笔者想与其如此，不如自己去检查内存。
下面介绍一下做法。
首先，暂时让486以后的CPU的高速缓存（cache）功能无效。
回忆一下最初讲的CPU与内存的关系吧。
我们说过，内存与CPU的距离地与CPU内部元件要远得多，因此在寄存器内部MOV，要比从寄存器MOV到内存快得多。
但另一方面，有一个问题，CPU的记忆力太差了，即使知道内存的速度不行，还不得不频繁使用内存。
考虑到这个问题，英特尔的大叔们在CPU里也加进了一点存储器，它被称为高速缓冲存储器（cache memory）。
cache这个词原是指储存粮食弹药等物资的仓库。
但是能够跟得上CPU速度的高速存储器价格特别高，一个芯片就有一个CPU那么贵。
如果128MB内存全部都用这种高价存储器，预算上肯定受不了。
高速缓存，容量只有这个数值的千分之一，也就是128KB左右。
高级CPU，也许能有1MB高速缓存，但即便这样，也不过就是128MB的分之一。
为了有效使用如此稀有的高速缓存，英特尔的大叔们决定，每次访问内存，都要将所访问的地址和内容存人到高速缓存里。
也就是存放成这样：18号地址的值是54。
如果下次再要用18号地址的内容，CPU就不再读内存了，而是使用高速缓存的信息，马上就能回答出l8号地址的内容是54。
往内存里写人数据时也一样，首先更新高速缓存的信息，然后再写入内存。
如果先写入内存的话，在等待写入完成的期间，CPU处于空闲状态，这样就会影响速度。
所以，先更新缓存，缓存控制电路配合内存的速度，然后再慢慢发送内存写入命令。

<<30天自制操作系统>>

编辑推荐

39.1K迷你系统，实现多任务、汉字显示、文件压缩，还能听歌看图玩游戏。
拥有《30天自制操作系统》一书，只需30天，从零开始编写一个五脏俱全的图形操作系统将不再只是个梦想。
　日本编程天才川合秀实，将通过本书，揭开CPU、内存、磁盘以及操作系统底层工作模式的神秘面纱。

<<30天自制操作系统>>

版权说明