

<<自己动手写操作系统>>

图书基本信息

书名：<<自己动手写操作系统>>

13位ISBN编号：9787121015779

10位ISBN编号：7121015773

出版时间：2005-8

出版时间：电子工业出版社

作者：于渊

页数：374

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<自己动手写操作系统>>

前言

你是否有过这样的经历，有一天你兴致勃勃买来一堆菜谱想学厨艺，翻开之后却发现自己根本没见过那些材料的名字，也不知道什么叫文火什么叫武火，什么叫上浆什么叫勾芡。

而菜谱里根本没告诉你！

你扔掉菜谱，垂头丧气，从此对厨艺失去兴趣。

你也可能会有这样的经历，当你在计算机课上学完了一堆C语言语法，想要大展身手实践一番的时候，突然发现你居然不知道源代码应该敲到哪里，是Word还是NotePad？

很多计算机自学者可能有过这样的经历，由于不知道如何跟踪调试，在辛辛苦苦编写的程序得不出正确的结果时，要么束手无策，要么用打印语句输出很多东西，费时费力，而教科书根本没教你这些操作的细节。

<<自己动手写操作系统>>

内容概要

本书在详细分析操作系统原理的基础上，用丰富的实例代码，一步一步地指导读者用C语言和汇编语言编写出一个具备操作系统基本功能的操作系统框架。
本书不同于其他的理论型书籍，而是提供给读者一个动手实践的路线图。
书中讲解了大量在开发操作系统中需注意的细节问题，这些细节不仅能使读者更深刻地认识操作系统的核心原理，而且使整个开发过程少走弯路。
全书共分7章。

本书适合各类程序员、程序开发爱好者阅读，也可作为高等院校操作系统课程的实践参考书。

<<自己动手写操作系统>>

书籍目录

第1章 马上动手写一个最小的“操作系统” 11.1 准备工作11.2 10分钟完成的操作系统11.3 Boot Sector31.4 代码解释31.5 水面下的冰山51.6 回顾6第2章 搭建你的工作环境72.1 虚拟计算机 (Virtual PC) 72.1.1 Virtual PC初体验82.1.2 创建你的第一个Virtual PC92.1.3 虚拟软盘研究122.1.4 虚拟软盘实战142.2 编译器 (NASM & GCC) 182.3 安装虚拟Linux192.4 在虚拟Linux上访问Windows文件夹262.5 安装虚拟PCDOS262.6 其他要素292.7 Bochs292.7.1 Bochs vs. Virtual PC vs. VMware302.7.2 Bochs的使用方法312.7.3 用Bochs进行调试332.7.4 在Linux上开发342.8 总结与回顾36第3章 保护模式 (Protect Mode) 373.1 认识保护模式373.1.1 GDT(Global Descriptor Table) 423.1.2 实模式到保护模式, 不一般的jmp453.1.3 描述符属性473.2 保护模式进阶503.2.1 海阔凭鱼跃503.2.2 LDT (Local Descriptor Table) 583.2.3 特权级623.3 页式存储823.3.1 分页机制概述833.3.2 编写代码启动分页机制843.3.3 PDE和PTE853.3.4 cr3883.3.5 回头看代码883.3.6 克勤克俭用内存903.3.7 进一步体会分页机制1003.4 中断和异常1073.4.1 中断和异常机制1093.4.2 外部中断1113.4.3 编程操作8259A1133.4.4 建立IDT1163.4.5 实现一个中断1173.4.6 时钟中断试验1193.4.7 几点额外说明1213.5 保护模式下的I/O1223.5.1 IOPL1223.5.2 I/O许可位图 (I/O Permission Bitmap) 1233.6 保护模式小结123第4章 让操作系统走进保护模式1254.1 突破512字节的限制1254.1.1 FAT121264.1.2 DOS可以识别的引导盘1314.1.3 一个最简单的Loader1324.1.4 加载Loader入内存1334.1.5 向Loader交出控制权1424.1.6 整理boot.asm1424.2 保护模式下的“操作系统” 144第5章 内核雏形1465.1 用NASM在Linux下写Hello World1465.2 再进一步, 汇编和C同步使用1485.3 ELF (Executable and Linkable Format) 1505.4 从Loader到内核1555.4.1 用Loader加载ELF1555.4.2 跳入保护模式1615.4.3 重新放置内核1705.4.4 向内核交出控制权1755.4.5 操作系统的调试方法1765.5 扩充内核1845.5.1 切换堆栈和GDT1845.5.2 整理我们的文件夹1915.5.3 Makefile1915.5.4 添加中断处理2005.5.5 两点说明2185.6 小结219第6章 进程2216.1 迟到的进程2216.2 概述2226.2.1 进程介绍2226.2.2 未雨绸缪——形成进程的必要考虑2226.2.3 参考的代码2246.3 最简单的进程2246.3.1 简单进程的关键技术预测2256.3.2 第一步——ring0 ring12276.3.3 第二步——丰富中断处理程序2436.3.4 进程体设计技巧2546.4 多进程2566.4.1 添加一个进程体2566.4.2 相关的变量和宏2576.4.3 进程表初始化代码扩充2586.4.4 LDT2606.4.5 修改中断处理程序2616.4.6 添加一个任务的步骤总结2636.4.7 号外: Minix的中断处理2656.4.8 代码回顾与整理2696.5 系统调用2806.5.1 实现一个简单的系统调用2806.5.2 get_ticks的应用2866.6 进程调度2926.6.1 避免对称——进程的节奏感2926.6.2 优先级调度总结300第7章 输入/输出系统3027.1 键盘3027.1.1 从中断开始——键盘初体验3027.1.2 AT、PS/2键盘3047.1.3 键盘敲击的过程3047.1.4 解析扫描码3097.2 显示器3257.2.1 初识TTY3257.2.2 基本概念3267.2.3 寄存器3287.3 TTY任务3327.3.1 TTY任务框架的搭建3347.3.2 多控制台3407.3.3 完善键盘处理3467.3.4 TTY任务总结3547.4 区分任务和用户进程3547.5 printf3577.5.1 为进程指定TTY3577.5.2 printf()的实现3587.5.3 系统调用write()3617.5.4 使用printf()363后记366参考文献369附录书中的章节和代码对照表370

<<自己动手写操作系统>>

章节摘录

前言 一年多以前，电子工业出版社的张毅编辑告诉我说，有一位年轻的程序员，正在写一本《自己动手写操作系统》的书。

知道这个消息，我既有点好奇，又有些担忧。

如果是在十年前，这样题材的书将会是读者争相传阅的对象，毕竟20世纪90年代是软件的理想主义年代。

但是在理想褪尽、实务未兴的尴尬的这两年，这样一本书在市场上究竟会遇到怎样的待遇，确实让人不敢乐观。

不过，在阅读了样章之后，我深为作者清新的文笔、流畅的思路和扎实的技术功底所折服，于是请张毅为我引见了这位作者，即本书的作者于渊。

于渊非常年轻，却有着高人一筹的表达能力和技术视野，我觉得他是难得的技术写作人才，就鼓励他在《程序员》杂志开辟了一个技术专栏，专门剖析操作系统相关的技术。

一年来这个专栏陆续发表了一系列文章，获得了不少读者的正面反馈。

然而，事实证明，我最初的担忧并不是没有道理的。

一年多来，不断有人表达过他们对这样一个题材的不同看法。

他们认为，相对于90年代中后期，现在的软件产业已经务实了很多，今天的程序员更关心的是如何尽可能快、尽可能简单地用软件解决实际问题，创造实际价值，在一个既定的秩序中寻找自己的生存空间，而不是异想天开地憧憬能成为Linus Torvalds式的旧秩序的“破坏者”。

因此诸如软件过程、开发方法、系统集成、应用架构等“高级”的话题受到关注和欢迎，而诸如操作系统、编译原理之类的基础技术，已经是关心者寥寥了。

他们非常怀疑，这样的一本书，对于一线的开发是否有实际的意义？

对于尚在寻找自己职业发展方向的初学者是否构成一种误导？

这个问题相当尖锐，必须面对。

我想这样一本书，至少在以下几个方面是具有重要的正面意义的。

首先，对于正在大学里学习计算机科学的学生来说，“操作系统原理”是重要的专业基础课。

为了达到大学阶段教育的标准，这方面的知识应当认真学习。

一些比较严肃的学校鼓励学生在学习这门课程的同时自己动手开发一个具体而微的操作系统。

这种实习对于学生充分掌握书本知识、打下扎实的基本功有非常大的好处。

在我认识的比较有成就的开发者中，有不少人自己动手写过小的操作系统，他们认为编写操作系统的实践使他们最终消除了对编写软件系统的心理障碍，实在地消化和理解了书本上的知识，学会了解决问题的思路，收获非常巨大。

可惜的是，大部分的学生都没有进行过这样的实践，这主要是因为目前的课本偏重操作系统理论，把大量的笔墨放在对操作系统运行机制的剖析或者现成源代码的分析上，对于那些想自己动手写一个操作系统的同学来说，从课本上反而得不到实际的指导。

即使是一些世界级的名著，在“How”上也是语焉不详。

在这方面，我相信于渊的这本书在国内算是填补了一个空白。

这本书最大的特点是明白、实在，将学习编写操作系统的每一个步骤都清清楚楚地交代出来，丝毫没有含糊其辞之处。

可以说，只要读者能够耐心阅读学习，按照书上交代的步骤一步步来，就肯定能够进入操作系统的大门，把书本上的知识与实践紧密联系起来。

毕竟写自己的操作系统是一个让所有程序员心动的事情。

如果当年我学习操作系统知识的时候能够有这样一本书，那该有多好！

其次，对于那些希望通过分析Linux源代码学习与研究操作系统，进而在开源软件天地里有所作为的研究者和开发者来说，这本书是非常好的入门阶梯。

目前研究Linux内核的图书，一般局限在对现有内核源代码的分析上，不但理解起来很困难，而且没有给读者以自己实践的机会。

<<自己动手写操作系统>>

有人想到去分析Linux早期的版本，降低了读者理解的难度，但是总的来说还是纸上谈兵。

本书的风格截然不同，不但行文活泼清新，叙理简明清晰，而且完全着眼于动手，以一种夹叙夹议的方式，对于编写操作系统过程中可能遇到的各种问题“逢山开路，遇水架桥”，读者可以在实际的语境中理解问题，解决问题。

通过这种方式学习操作系统的实现技术，无疑要比其他方式更为有效。

而且，于渊在这本书中构造的这个微型的操作系统，跟Linux有微妙的相关，读者细心品味便知。

另外，虽然目前国内软件产业的主流是做下游的生产性集成，但是对于程序员个体来说，也有不少从事系统级软件开发的机会。

有幸从事系统级软件开发的朋友，更是可以直接地从本书中学到不少实用的知识和技能。

特别是作者在解决一个又一个问题的过程中所体现出来的思路和方法，可能是更值得大家学习的东西。

众所周知，操作系统是计算机软件领域中核心的工程性技术，尽管它的理论相对成熟，但是在工程实施和维护上，仍然是体现一个国家软件技术水平的“两弹、一星、大飞机”级的标志性核心技术。

世界上凡是在软件产业方面存有雄心壮志的国家，无不非常重视操作系统技术的研究和积累。

比如法国在他们的一个国家级实验室中，自己研发了包括操作系统和编译器在内的全套基础软件，并由国家投入资金不断维护和发展。

德国拥有大批Linux黑客，其政府因势利导，通过一系列的大型工程将自己的Linux软件人才组织起来，希望依托Linux重建自己的软件核心技术力量。

20世纪80年代中期，日本在美国的压力下而放弃了自己的“BTRON”操作系统，此后软件产业的发展让日本追悔莫及。

痛定思痛之后，日本希望牢牢地把握自己在消费电子产品上的优势，一方面继续发展国产的ITRON OS，另一方面把握住Linux的机会，希望在未来占据消费类嵌入式操作系统的制高点。

我国在这个方面走过一些弯路，但是现在已经认识到了掌握核心软件技术的重要性，并且有了一定的投入，相信今后国家在这方面的支持力度会越来越强。

我本人见过国内的一些操作系统方面的专家，切实地感到，就个体而言，国内的技术专家在理论和实践上都达到了很高的水平，但是由于缺乏一个质量高、并且有一定规模的团队和社群，他们基本处于单打独斗或者小组作战的状态，不仅个人的技术不能够得到充分地发挥，而且也不能形成有规模成果，无法从根本上扭转我国在软件核心技术领域上的劣势。

我认为，只有在中国出现一大批关心操作系统、熟悉操作系统的程序员，才有可能逐渐缩小我们与世界先进水平的差异。

作为中国软件产业中的普通一员，我非常希望看到这本书能够在这个过程中发挥一点作用。

孟岩《程序员》杂志技术主编 2005年7月

<<自己动手写操作系统>>

媒体关注与评论

这是一本编程爱好者编写的别具一格、颇有特色的操作系统原理与实现的书。该书作者对操作系统具有特殊爱好，在大量实践和反复钻研下积累了丰富而可贵的经验，为了与广大读者分享这些经验写成了此书。

本书对一般的操作系统原理教材不很重视的部分，例如，系统初启、保护模式、控制权如何转入OS Kernal等都写得具体详细，对操作系统的爱好者以及涉足于操作系统设计、实现和应用的读者有很好的参考价值。

本书的文字生动活泼，富有个性，可望提高青年学子的阅读兴趣。

<<自己动手写操作系统>>

编辑推荐

《自己动手写操作系统》是一本编程爱好者编写的别具一格、颇有特色的操作系统原理与实现的书。

该书作者对操作系统具有特殊爱好，在大量实践和反复钻研下积累了丰富而可贵的经验，为了与广大读者分享这些经验写成了此书。

本书对一般的操作系统原理教材不很重视的部分，例如，系统初启、保护模式、控制权如何转入OS Kernal等都写得具体详细，对操作系统的爱好者以及涉足于操作系统设计、实现和应用的读者有很好的参考价值。

《自己动手写操作系统》的文字生动活泼，富有个性，可望提高青年学子的阅读兴趣。

《自己动手写操作系统》在详细分析操作系统原理的基础上，用丰富的实例代码，一步一步地指导读者用C语言和汇编语言编写出一个具备操作系统基本功能的操作系统框架。

本书不同于其他的理论型书籍，而是提供给读者一个动手实践的路线图。

书中讲解了大量在开发操作系统中需注意的细节问题，这些细节不仅能使读者更深刻地认识操作系统的核心原理，而且使整个开发过程少走弯路。

全书共分7章。

《自己动手写操作系统》适合各类程序员、程序开发爱好者阅读，也可作为高等院校操作系统课程的实践参考书。

<<自己动手写操作系统>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>