

图书基本信息

书名：<<Linux内核设计的艺术图解Linux操作系统架构设与实现原理>>

13位ISBN编号：9787111421764

10位ISBN编号：7111421760

出版时间：2013-5

出版时间：新设计团队 机械工业出版社 (2013-05出版)

作者：新设计团队

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>

前言

【前言】为什么写这本书很早就有一个想法，做中国人自己的、有所突破、有所创新的操作系统、计算机语言及编译平台。

我带领的“新设计团队”（主要由中国科学院研究生院毕业的学生组成）在实际开发自己的操作系统的过程中，最先遇到的问题就是如何培养学生真正看懂Linux操作系统的源代码的能力。

开源的Linux操作系统的源代码很容易找到，但很快就会发现，培养学生看懂Linux操作系统的源代码是一件非常困难的事。

操作系统的代码量通常都是非常庞大的，动辄几百万行，即使浏览一遍也要很长时间。

比庞大的代码量更让学习者绝望的是操作系统有着极其错综复杂的关系。

看上去，代码的执行序时隐时现，很难抓住脉络。

代码之间相互牵扯，相互勾连，几乎无法理出头绪，更谈不上理解代码背后的原理、意图和思想。

对于学生而言，选择从源代码的什么地方开始分析，本身就是一个难题。

通常，学生有两种选择：一种是从main函数，也就是从C语言代码的总入口开始，沿着源代码的调用路线一行一行地看下去，学生很快就会发现源代码的调用路线莫名其妙地断了，但直觉和常识告诉他操作系统肯定不会在这个地方停止，一定还在继续运行，却不知道后续的代码在哪里，这种方法很快就走进了死胡同；另一种则是从某一模块入手，如文件系统，但这样会无形中切断操作系统源码之间复杂的关系，如文件系统与进程管理的关系，文件系统与内存管理的关系，等等。

学生如果孤立地去理解一个模块，往往只能记住一些名词和简单概念，难以真正理解操作系统的全貌。

用学生的话讲，他们理解的操作系统变成了“文科”的操作系统。

由于操作系统是底层系统程序，对应用程序行之有效的调试和跟踪等手段对操作系统的源代码而言，几乎无效。

学生就算把每一行源代码都看懂了，对源代码已经烂熟于心，知道这一行是一个for循环，那一行是一个调用……但仍然不知道整个代码究竟在做什么以及起到什么作用，更不知道设计者的意图究竟是什么。

学生在操作系统课程上学习过进程管理、内存管理、文件系统等基础知识，但是对这些空洞的理论在一个实际的操作系统中是如何实现的却不得而知。

他们在源代码中很难看出进程和内存之间有什么关联，内核程序 and 用户程序有什么区别，为什么要有这些区别；也很难从源代码中看清楚，我们实际经常用到的操作，比如打开文件，操作系统在其中都做了哪些具体的工作。

想在与常见的应用程序的编程方法有巨大差异的、晦涩难懂的、浩瀚如海的操作系统底层源代码中找到这些问题的答案，似乎比登天还难。

对熟悉操作系统源代码的学生而言，他们也知道像分页机制这样的知识点，知道若干级的分页及恒等映射，但是未必能够真正理解隐藏在机制背后的深刻意义。

这些都是学生在学习Linux操作系统源代码时遇到的实际问题。

中国科学院研究生院的学生应该是年轻人中的佼佼者，他们遇到的问题可能其他读者也会遇到。

我萌发了一个想法，虽然学生的问题早已解决，但是否可以把他们曾经在学习、研发操作系统的过程中遇到的问题和心得体会拿出来供广大读者分享。

当时，针对学生的实际问题，我的解决方法是以一个真实的操作系统为例，让学生理解源代码并把操作系统在内存中的运行时状态画出图来。

实践证明，这个方法简单有效。

现在我们把这个解决方案体现在这本书中。

这就是以一个真实的操作系统的实际运行为主线；以图形、图像为核心，突出描述操作系统在实际运行过程中内存的运行时结构；强调学生站在操作系统设计者的视角，用体系的思想方法，整体把握操作系统的行为、作用、目的和意义。

第1版与第2版的区别第2版较第1版有较大的改动。

从总体结构上，将第1版的第2章拆分为第2版的第2章、第3章、第4章。这样的拆分对操作系统启动部分的系统初始化、激活进程0、创建进程1、创建进程2的层次划分更清晰。

各章内容的分量也比较均衡，阅读感受会更好。

根据读者的反馈意见，第2版增加了一些示意图，在源代码中增加了大量的注释，对操作系统的架构表述得更直观，对源代码讲解得更细致。

这些是第2版改动最大、下功夫最多的地方。

希望我们的努力能给读者带来更多的帮助。

本书内容及特色在全书的讲解过程中，我们不仅详细分析了源代码、分析了操作系统的执行序，还特别分析了操作系统都做了哪些“事”，并且对于“事”与“事”之间的关系和来龙去脉，这些“事”意味着什么，为什么要做这些“事”，这些“事”背后的设计思想是什么……都做了非常详细且深入的分析。

更重要的是，对于所有重要的阶段，我们几乎都用图解的方式把操作系统在内存中的实际运行状态精确地表示了出来。

我们用600 dpi的分辨率精心绘制了300多张图，图中表现的运行时结构和状态与操作系统实际运行的真实状态完全吻合。

对每一条线、每一个色块、每一个位置、每一个地址及每一个数字，我们都经过了认真反复地推演和求证，并最终在计算机上进行了核对和验证。

看了这些绘制精美的图后，读者的头脑中就不再是一行行、一段段枯燥的、令人眩晕的源代码，而是立体呈现的一件件清晰的“事”，以及这些“事”在内存中直截了当、清晰鲜活的画面。

用这样的方法讲解操作系统是本书的一大特色。

理解这些图要比理解源代码和文字容易得多。

毫不夸张地说，只要你能理解这些图，你就理解了操作系统的80%。

这时你可以自豪地说，你比大多数用别的方法学过操作系统的人的水平都要高出一大截。

作者和机械工业出版社的编辑做了大量的检索工作。

就我们检索的范围而言，这样的创作方法及具有这样特色的操作系统专著在世界范围都是第一次。

本书分三部分来讲解Linux操作系统：第一部分（第1~4章）分析了从开机加电到操作系统启动完成并进入怠速状态的整个过程；第二部分（第5~8章）讲述了操作系统进入系统怠速后，在执行用户程序的过程中，操作系统和用户进程的实际运行过程和状态；第三部分（第9章）阐述整个Linux操作系统的设计指导思想，是从微观到宏观的回归。

第一部分中，我们详细讲解了开机加电启动BIOS，通过BIOS加载操作系统程序，对主机的初始化，打开保护模式和分页，调用main函数，创建进程0、进程1、进程2以及shell进程，并且具备用文件的形式与外设交互。

第二部分中，我们设计了几个尽可能简单又有代表性的应用程序，并以这些程序的执行为引导，详细讲解了安装文件系统、文件操作、用户进程与内存管理、多个进程对文件的操作以及进程间通信。

我们将操作系统的原理自然而然地融入了讲解真实操作系统的实际运行过程中。

在读者看来，操作系统原理不再是空对空的、“文科”概念的计算机理论，而是既有完整且体系的理论，又有真实、具体、实际的代码和案例，理论与实际紧密结合。

第三部分是全书水平最高的部分，详细阐述了主奴机制以及实现主奴机制的三项关键技术：保护和分页、特权级、中断，分析了保障主奴机制实现的决定性因素—先机，还详细讲解了缓冲区、共享页面、信号、管道的设计指导思想。

我们尝试从操作系统设计者的视角讲解操作系统的设计指导思想。

希望帮助读者用体系的思想理解、把握、驾驭整个操作系统以及背后的设计思想和设计意图。

在本书中，我们详细讲解了大家在学习操作系统的过程中可能会遇到的每一个难点，如main函数中的pause()调用，虽然已经找不到后续代码，但该调用结束后，程序仍然执行的原因是：中断已经打开，进程调度就开始了，而此时可以调度的进程只有进程1，所以后续的代码应该从进程1处继续执行……我们还对读者不容易理解和掌握的操作系统特有的底层代码的一些编程技巧做了详细的讲解，如用

模拟call的方法，通过ret指令“调用”main函数……总之，我们所做的一切努力就是想真正解决读者遇到的实际问题和难题，给予读者有效的帮助。

我们盼望即使是刚刚考入大学的学生也有兴趣和信心把这本书读下去；我们同样希望即使是对操作系统源代码很熟悉的读者，这本书也能给他们一些不同的视角、方法和体系性思考。

为什么本书选用Linux 0.11内核这本书选用的是Linux 0.11操作系统源代码。

对为什么选用Linux 0.11而不是最新版本，赵炯先生有过非常精彩的论述。

我们认为赵先生的论述是非常到位的。

我们不妨看一下Linux最新的版本2.6，代码量大约在千万行这个量级，去掉其中的驱动部分，代码量仍在百万行这个量级。

一个人一秒钟看一行，一天看8小时，中间不吃、不喝、不休息，也要看上几个月，很难想象如何去理解。

就算我们坚持要选用Linux 2.6，就算我们写上2000页（书足足会有十几厘米厚），所有的篇幅都用来印代码，也只能印上不到十分之一的代码。

所以，即使是这么不切实际的篇幅，也不可能整体讲解Linux 2.6。

读者会逐渐明白，对于理解和掌握操作系统而言，真正有价值的是整体、是体系，而不是局部。

Linux 0.11的内核代码虽然只有约两万行，但却是一个实实在在、不折不扣的现代操作系统。

因为它具有现代操作系统最重要的特征—支持实时多任务，所以必然支持保护和分页……而且它还是后续版本的真正的始祖，有着内在的、紧密的传承关系。

读者更容易看清设计者最初的、最根本的设计意图和设计指导思想。

Linux 0.11已经问世20多年了，被世人广为研究和学习。

换一个角度看，要想对众人熟悉的事物和领域讲出新意和特色，对作者来说也是一个强有力的挑战。

致谢首先，感谢机械工业出版社华章公司的副总经理温莉芳女士以及其他领导，是他们的决心和决策成就了这本书，并且在几乎所有方面给予了强有力的支持。

特别令人感动的是他们主动承担了全部的出版风险，同时给予了作者最好的条件，让我们看到一个大出版社的气度和风范。

其次，特别感谢机械工业出版社华章公司的编辑杨福川。

杨先生的鉴赏力和他的事业心以及他对工作认真负责的态度为这本书的出版打开了大门。

杨先生对读者的理解以及他的计算机专业素养使得他有能力对这本书给予全方位的指导和帮助，使我们对这本书整体修改了多次，使之更贴近读者，可读性更好。

还要感谢我们和杨福川共同的朋友张国强先生和杨缙女士。

最后，感谢我们的家人和朋友，是他们坚定的支持才使得整个团队能够拒绝方方面面、形形色色的诱惑，放弃普遍追求的短期利益；我们在常人难以想象的艰苦条件下，长时间专注于操作系统、计算机语言、编译器、计算机体系结构等基础性学科的研究。

因为我们认认真真、踏踏实实、不为名利，只为做一点实在、深入的工作，积累了十年的经验，打造了一支敢想、敢干、敢打、敢拼、不惧世界顶级强敌的队伍。

这些是本书的基础。

杨力祥中国科学院研究生院2013年1月

内容概要

《Linux内核设计的艺术图解Linux操作系统架构设与实现原理(第2版)》的第1版广受好评，版权被中国台湾和美国两家大型出版社引进，第2版根据读者的反馈和作者对操作系统的最新研究成果对第1版进行了大幅优化和重写，使其内容质量更上一层楼。

《Linux内核设计的艺术图解Linux操作系统架构设与实现原理(第2版)》在众多关于Linux内核的书中独树一帜，它在世界范围内首次提出并阐述了操作系统设计的核心指导思想——主奴机制，这是所有操作系统研究者的一笔宝贵财富。

它也是一本能真正引导我们较为容易地、极为透彻地理解Linux内核的经典之作，也可能是当前唯一能从本质上指引我们去设计和开发拥有自主知识产权的操作系统的著作。

《Linux内核设计的艺术图解Linux操作系统架构设与实现原理(第2版)》的最大特点是它的写作方式和内容组织方式与同类书完全不同。

它在深刻地分析了传统讲解方法的利弊之后，破旧立新，从认知学的角度开创了一种全新的方式。以操作系统的真实运行过程为主线，结合真实的内核源代码、300余幅精确的内核运行时序图和具有点睛之妙的文字说明，对操作系统从开机加电到系统完全准备就绪，及运行用户程序的整个过程进行了系统而完整地分析，深刻地揭示了其间每一个动作的设计意图和实现原理，完美地再现了操作系统设计者的设计思路。

阅读《Linux内核设计的艺术图解Linux操作系统架构设与实现原理(第2版)》就如同跟随着操作系统设计者一起去思考，我们会在阅读的过程中发现Linux内核设计的精妙，会发现原来处处都“暗藏玄机”，哪怕是一行很短的代码。

《Linux内核设计的艺术图解Linux操作系统架构设与实现原理(第2版)》在所有细节上都力求完美。为了保证知识的准确性，操作系统运行过程中的每个动作都经过了严格的考证；为了让我们真正理解Linux内核的原理，它突破传统，以Linux的真实运行过程为主线进行讲解；为了做到真正易于理解，创新性地使用了图解的方式，精心绘制了300余幅分辨率600dpi的时序图，图中表现的运行时结构和状态与操作系统实际运行时的真实状态完全吻合；为了提高阅读体验，《Linux内核设计的艺术图解Linux操作系统架构设与实现原理(第2版)》采用了双色印刷，以便于我们更清楚地观察每一幅图中的细节。

作者简介

长期以来，新设计团队一直在为设计一个自主的、有所突破和创新的操作系统而努力。为了让新的成员能更快、更容易地理解操作系统的精髓，从更高的角度去鉴赏和发现操作系统设计中的精妙与不足，团队成员以Linux 0.11内核为例，对操作系统的设计思想和实现原理进行了深刻地剖析，取得了十分好的效果，很好地培养和锻炼了团队成员对操作系统的驾驭能力。为了实现让国人也能设计出自己的目标，本书作者团队无私地将他们的独特研究方式与研究成果奉献了出来，希望所有想要去深刻理解Linux内核和操作系统设计思想朋友能从中受益。

书籍目录

前言第1章 从开机加电到执行main函数之前的过程11.1 启动BIOS, 准备实模式下的中断向量表和中断服务程序11.1.1 BIOS的启动原理21.1.2 BIOS 在内存中加载中断向量表和中断服务程序31.2 加载操作系统内核程序并为保护模式做准备51.2.1 加载第一部分内核代码——引导程序 (bootsect) 51.2.2 加载第二部分内核代码——setup71.2.3 加载第三部分内核代码——system模块131.3 开始向32位模式转变, 为main函数的调用做准备171.3.1 关中断并将system移动到内存地址起始位置0x00000171.3.2 设置中断描述符表和全局描述符表191.3.3 打开A20, 实现32位寻址211.3.4 为保护模式下执行head.s做准备231.3.5 head.s开始执行261.4 本章小结44第2章 设备环境初始化及激活进程0452.1 设置根设备、硬盘462.2 规划物理内存格局, 设置缓冲区、虚拟盘、主内存462.3 设置虚拟盘空间并初始化482.4 内存管理结构mem_map初始化502.5 异常处理类中断服务程序挂接512.6 初始化块设备请求项结构572.7 与建立人机交互界面相关的外设的中断服务程序挂接592.7.1 对串行口进行设置592.7.2 对显示器进行设置602.7.3 对键盘进行设置612.8 开机启动时间设置632.9 初始化进程0652.9.1 初始化进程0682.9.2 设置时钟中断712.9.3 设置系统调用总入口712.10 初始化缓冲区管理结构732.11 初始化硬盘752.12 初始化软盘772.13 开启中断782.14 进程0由0特权级翻转到3特权级, 成为真正的进程782.15 本章小结80第3章 进程1的创建及执行813.1 进程1的创建813.1.1 进程0创建进程1813.1.2 在task[64]中为进程1申请一个空闲位置并获取进程号873.1.3 调用copy_process函数893.1.4 设置进程1的分页管理943.1.5 进程1共享进程0的文件993.1.6 设置进程1在GDT中的表项993.1.7 进程1处于就绪态1003.2 内核第一次做进程调度1033.3 轮转到进程1执行1073.3.1 进程1为安装硬盘文件系统做准备1093.3.2 进程1格式化虚拟盘并更换根设备为虚拟盘1353.3.3 进程1在根设备上加载根文件系统1383.4 本章小结151第4章 进程2的创建及执行1524.1 打开终端设备文件及复制文件句柄1524.1.1 打开标准输入设备文件1524.1.2 打开标准输出、标准错误输出设备文件1664.2 进程1创建进程2并切换到进程2执行1694.3 加载shell程序1784.3.1 关闭标准输入设备文件, 打开rc文件1784.3.2 检测shell文件1814.3.3 为shell程序的执行做准备1864.3.4 执行shell程序1924.4 系统实现怠速1964.4.1 创建update进程1964.4.2 切换到shell进程执行1984.4.3 重建shell2044.5 本章小结205第5章 文件操作2065.1 安装文件系统2065.1.1 获取外设的超级块2065.1.2 确定根文件系统的挂接点2095.1.3 将超级块与根文件系统挂接2105.2 打开文件2115.2.1 将进程的*filp[20]与file_table[64]挂接2125.2.2 获取文件i节点2135.2.3 将文件i节点与file_table[64]挂接2235.3 读文件2245.3.1 确定数据块在外设中的位置2245.3.2 将数据块读入缓冲块2285.3.3 将缓冲块中的数据复制到进程空间2285.4 新建文件2305.4.1 查找文件2305.4.2 新建文件i节点2315.4.3 新建文件目录项2335.5 写文件2385.5.1 确定文件的写入位置2385.5.2 申请缓冲块2415.5.3 将指定的数据从进程空间复制到缓冲块2415.5.4 数据同步到外设的两种方法2425.6 修改文件2455.6.1 重定位文件的当前操作指针2465.6.2 修改文件2465.7 关闭文件2485.7.1 当前进程的filp与file_table[64]脱钩2485.7.2 文件i节点被释放2495.8 删除文件2505.8.1 对文件的删除条件进行检查2515.8.2 进行具体的删除工作2525.9 本章小结255第6章 用户进程与内存管理2566.1 线性地址的保护2566.1.1 进程线性地址空间的格局2566.1.2 段基址、段限长、GDT、LDT、特权级2576.2 分页2606.2.1 线性地址映射到物理地址2606.2.2 进程执行时分页2616.2.3 进程共享页面2676.2.4 内核分页2706.3 一个用户进程从创建到退出的完整过程2736.3.1 创建str1进程2736.3.2 str1进程加载的准备工作2856.3.3 str1进程的运行、加载2896.3.4 str1进程的退出2966.4 多个用户进程同时运行2996.4.1 进程调度2996.4.2 页写保护3036.5 本章小结309第7章 缓冲区和多进程操作文件3107.1 缓冲区的作用3107.2 缓冲区的总体结构3117.3 b_dev、b_blocknr及request的作用3127.3.1 保证进程与缓冲块数据交互的正确性3127.3.2 让数据在缓冲区中停留的时间尽可能长3207.4 uptodate和dirty的作用3257.4.1 b_uptodate的作用3267.4.2 b_dirty的作用3317.4.3 i_uptodate、i_dirty和s_dirty的作用3347.5 count、lock、wait、request的作用3367.5.1 b_count的作用3367.5.2 i_count的作用3387.5.3 b_lock、*b_wait的作用3417.5.4 i_lock、i_wait、s_lock、*s_wait的作用3447.5.5 补充request的作用3477.6 实例1: 关于缓冲块的进程等待队列3497.7 总体来看缓冲块和请求项3707.8 实例2: 多进程操作文件的综合实例3737.9 本章小结388第8章 进程间通信3898.1 管道机制3898.1.1 管道的创建过程3918.1.2 管道的操作3968.2 信号机制4108.2.1 信号的使用4128.2.2 信号对进程执行状态的影响4228.3 本章小结431第9章 操作系统的设计指导思想4329.1 运行一个最简单的程序, 看操作系统为程序运行做了哪些工作4329.2 操作系统的设计指导思想——主奴机制4349.2.1 主奴机制中的进程及进程创建机制4359.2.2 操

作系统的设计如何体现主奴机制4369.3 实现主奴机制的三种关键技术4389.3.1 保护和分页4389.3.2 特权级4409.3.3 中断4419.4 建立主奴机制的决定性因素——先机4439.5 软件和硬件的关系4449.5.1 非用户进程——进程0、进程1、shell进程4449.5.2 文件与数据存储4459.6 父子进程共享页面4509.7 操作系统的全局中断与进程的局部中断——信号4509.8 本章小结451结束语452 “新设计团队”简介453

编辑推荐

《Linux内核设计的艺术图解Linux操作系统架构设计与实现原理(第2版)》编辑推荐：畅销书全新大幅升级，第1版广获好评，被翻译为繁体中文和英文出版，从操作系统设计者的视角，用体系的思想方法，深刻解读操作系统的架构设计与实现原理。

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>