HZ BOOKS
华章科技

本书对操作系统内核的驾驭能力和深刻理解程度达到世界顶尖级水平，是一本能真正引导你深入理解Linux内核设计思想的经典著作。

华章原创精品

Linux内核设计的艺术

图解Linux操作系统架构设计与实现原理

The Art of Linux Kernel Design

新设计团队 著

机械工业出版社
China Machine Press

华章原创精品

Linux内核设计的艺术

图解Linux操作系统架构设计与实现原理

The Art of Linux Kernel Design

新设计团队 著

机械工业出版社
China Machine Press

关于 Linux 内核的书已经不计其数，但这本书却是独树一帜的，它的内容代表着 Linux 内核研究成果的世界顶尖级水平，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。本书可能也代表着同类图书的顶尖水平，是一本真正能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。它的出版也许会成为 Linux 内核研究领域的一个里程碑事件。

本书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、349 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。阅读本书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。

本书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 349 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，本书采用了双色印刷，以便于我们更清楚地观察每一幅图中的细节。

图书在版编目（CIP）数据

Linux 内核设计的艺术：图解 Linux 操作系统架构设计与实现原理 / 新设计团队著 . —北京：机械工业出版社，2011.5

ISBN 978-7-111-34744-6

Ⅰ . L… Ⅱ . 新… Ⅲ . Linux 操作系统 – 图解 Ⅳ . TP316-64

中国版本图书馆 CIP 数据核字（2011）第 088008 号

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）
责任编辑：陈佳媛
北京京师印务有限公司印刷
2011 年 7 月第 1 版第 2 次印刷
186mm×240mm • 27.75 印张
标准书号：ISBN 978-7-111-34744-6
定价：79.00 元

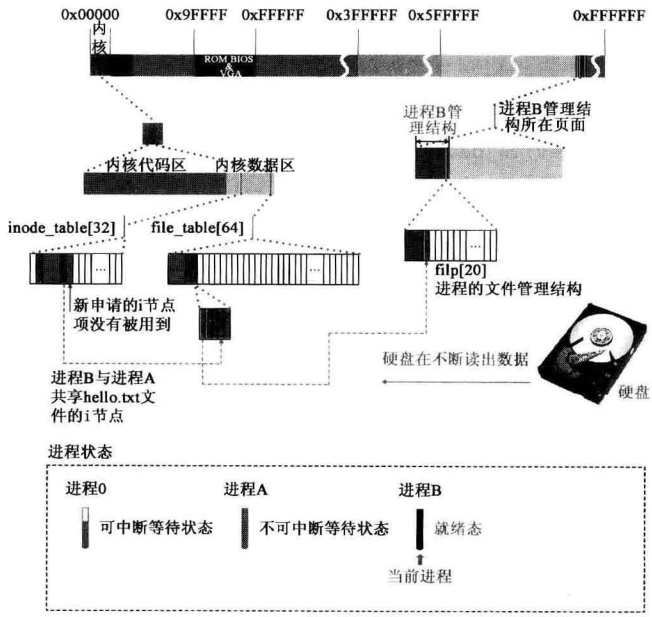
凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

本书导读

一、本书约定

1. 图的约定

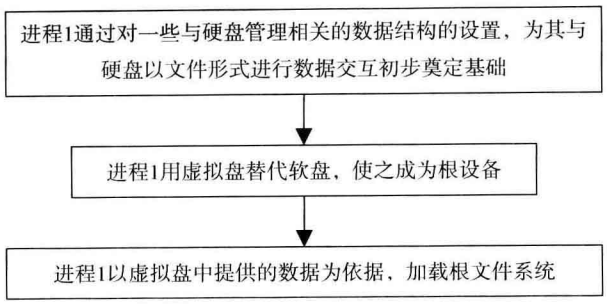
以本书中第 6 章中的图 6-5 为例来说明本书中关于图的约定。



这张图描绘了操作系统运行时的一个场景，最上面的一个横条表示的是内存，其上的 0x00000、0x9FFFF 等都是内存地址值。在内存示意图上，用颜色和灰阶表示了操作系统的内核、缓冲区、虚拟盘区和主内存区。从内存示意图中通过虚线向下引出了更多的横条，它们是在对内存中的局部区域进行放大。这些局部区域就是这个场景中需要关注的内存的细节，它们表现了涉及的代码和数据在内存中的精确位置。虚线箭头表示数据的传递和挂接情况。我们在图中绘制了硬盘的状态，表示此时硬盘正在不断地读出数据。另外，在图的最下方，用三个小竖条表示了三个进程的状态，灰色表示等待状态，黑色表示就绪态，小竖条中的空白表示进程已用的时间片。

对于比较复杂的章节，为了能让读者在阅读之前对该章节的内容有一个总体的了解，我们绘制了程序架构图。以第 2 章 2.3 节加载根文件系统的程序架构图（如下所示）为例，这张图表示

该小节的内容分为三个阶段，框图中写出了每个阶段的主要意思。



2. 源代码的约定

以本书第 6 章 6.1.5 小节的内容所对应的源代码为例来说明本书对源代码的约定。

```
// 代码路径：fs/inode.c：
struct m_inode * iget(int dev,int nr)
{
    ......
    while (inode < NR_INODE+inode_table)
    {
        if (inode->i_dev != dev || inode->i_num != nr)
        ......
        inode->i_count++;
        ......
        if (empty)
            iput(empty);
        return inode;
    }
    ......
}
```

第 1 行是该部分源代码在 linux0.11 内核源代码中的路径，通过这个路径，读者可以很容易地找到相关的函数。由于篇幅有限，本书没有贴出全部的源代码，只选取了读者最需要关注的部分。

3.“小贴士”的约定

对于那些可能会影响读者阅读的术语，本书在它们第一次出现时用“小贴士”进行了简单的解释。例如：

小贴士

8259A 中断控制器：8259A 是专门为了对 8085A 和 8086/8088 进行中断控制而设计的芯片，它是可以用程序控制的中断控制器。单个的 8259A 能管理 8 级向量优先级中断。在不增加其他电路的情况下，最多可以级联成 64 级的向量优先级中断系统。

4.“点评”的约定

本书对一些深刻难懂、能反映操作系统精妙设计思想和需要读者特别注意的知识点进行了深度点评，以帮助读者更好地理解本书的内容。例如：

点评

为什么没有最先调用 main 函数？

学过 C 语言的都知道，用 C 语言设计的程序都有一个 main 函数，而且是从 main 函数开始执行的。linux 0.11 的代码是用 C 语言编写的，奇怪的是，为什么在操作系统启动时先执行的是三个由汇编写成的程序，然后才开始执行 main 函数？为什么不是像我们熟知的 C 程序那样，从 main 函数开始执行呢？

……

二、阅读建议

首先，强烈建议读者按照本书的目录顺序进行阅读！此外，还有四点需要注意：

第一，建议读者不要因为某一个地方没有看懂就停下来长时间思考，而应该继续阅读，到后续章节中去寻找答案。比如，第 2 章的内容几乎涉及了整个操作系统的重要理论知识，对于非重点的这些内容更深入细致的描述，我们都安排在了后续的章节中讲解。

第二，注意阅读的层次。对于比较复杂和难懂的章节，我们在开篇处对本章的主要思想和想要讲解的内容进行了宏观的介绍，而且还给出了直观的程序架构图。这部分内容虽然篇幅不大，但是对理解后续的内容非常重要，请读者特别注意。

第三，请一定将文字说明、图和源代码三者结合起来阅读。操作系统的设计思想和实现原理都蕴含在源代码中，需要读者去分析、推断、猜测和验证，不同水平的读者能看到和领悟到的内容会有所不同。文字能很好地对源代码进行辅助说明，而且能尽可能地详细，但是理解起来不够直接。图能很直观地展示一些核心的内容，但是仅凭图能读懂多少信息，这也与读者的水平有很大的关系。说明文字、图和源代码三者各有优势，而且能互相弥补彼此的不足，阅读时如果能有效地结合三者，一定事半功倍。

第四，操作系统毕竟是非常难的，如果有些内容一时没有看懂，这是很正常的。对于不太容易理解的内容，建议读者整章整节地反复阅读。操作系统最难和最有价值的是它的整体，整体看明白了，局部才能清楚。哪怕对整体只有一个朦胧的了解，对理解操作系统都是有意义的。

三、如何获得 Linux 0.11 源代码

Linux 0.11 的源代码是开源的，您可以从赵炯先生维护的网站 http://www.oldlinux.org/Linux.old/Linux-0.11/ 上获取本书所涉及的全部源代码。

四、搭建阅读本书的实践操作环境

我们强烈建议您一边阅读本书，一边对照着 Linux 的源代码进行实践，只有这样才能真正理解 Linux 操作系统。在开始之前，你需要搭建一个 Linux 0.11 系统环境，具体方法见附录 A。

前　言

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已毕业的学生组成）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，即使是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，但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。使学生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。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，以及起什么作用，更不知道设计者的意图究竟是什么。

我们在操作系统的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，内核程序和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知识点，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得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，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。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。

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，我们还特别分析了操作系统都作了哪些“事”，并且把“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 600dpi 的分辨率精心绘制了 349 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。每一条线、每一个色块、每一个位置、每一个地址及数字都经过了我们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，在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%，这时你可以自豪的说，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作了大量的检索工作，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。

我们分三个部分来讲解 Linux 操作系统：第一部分由第 1 章和第 2 章组成，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；第二部分由第 3 章、第 4 章、第 5 章、第 6 章、第 7 章组成，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；第三部分由第 8 章组成，阐述整个 Linux 操作系统的设计指导思想，本章内容是从微观到宏观的回归。

第一部分，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，并且具备以文件的形式与外设交互。

第二部分，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到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，而是既有完整的、体系的理论，又有真实、具体、实际的代码和案例，理论与实际紧密耦合。

第三部分是全书水平最高的部分，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。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。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：中断一经打开，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……

我们还对一些读者不容易理解和掌握的操作系统特有的底层代码的编程技巧作了详细的讲解。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。

这本书选用的操作系统源代码是 Linux 0.11。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。

就算我们硬要选用 Linux 2.6，就算我们写上 3000 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 年了，被世人广为研究和学习。换一个角度看，要想在众人熟悉的

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。

这本书能够顺利出版，我们首先要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，我们还要感谢的是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6次，使之更贴近读者，可读性更好。

我们还要感谢我们和杨福川先生共同的朋友张国强先生和杨缙女士。

最后，我们要感谢我们的家人和朋友。是他们坚定的支持才使得我们的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，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。认认真真、踏踏实实、不为名利，做了一点实在、深入的工作，有了一点能凑合拿得出手的10年积累，以及一支敢想、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

2011年5月

目　录