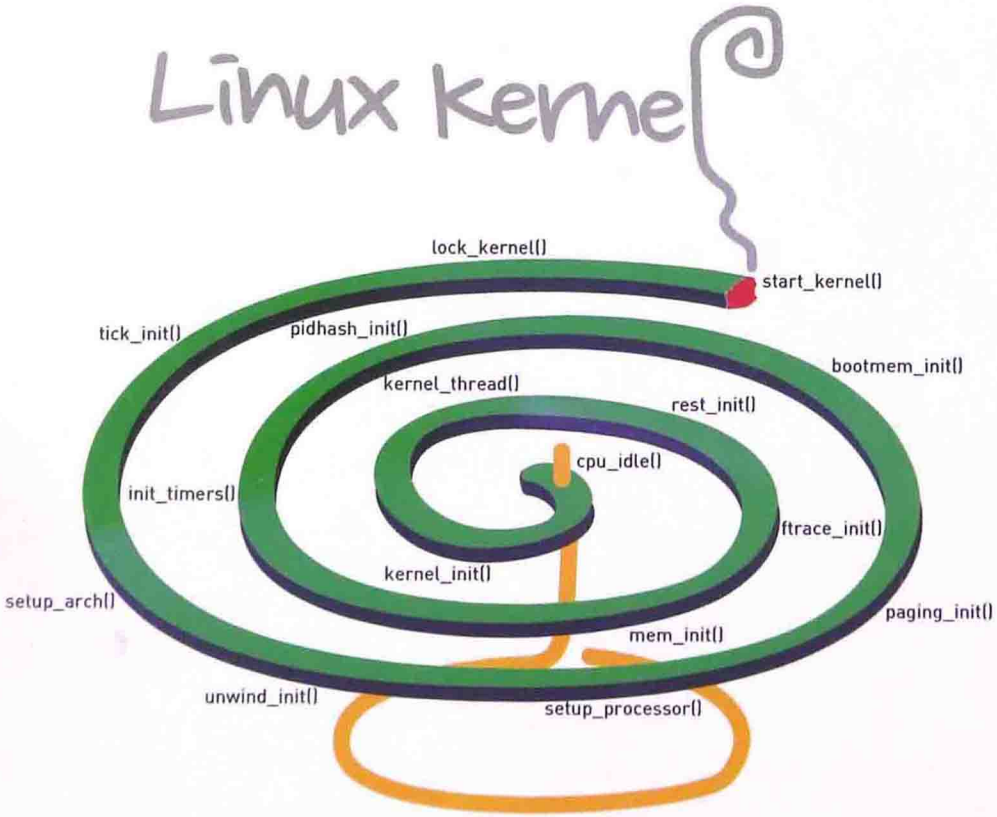
TURING 图灵程序设计丛书

ARM Linux 内核源码剖析

【韩】尹锡训等 著　崔范松 译



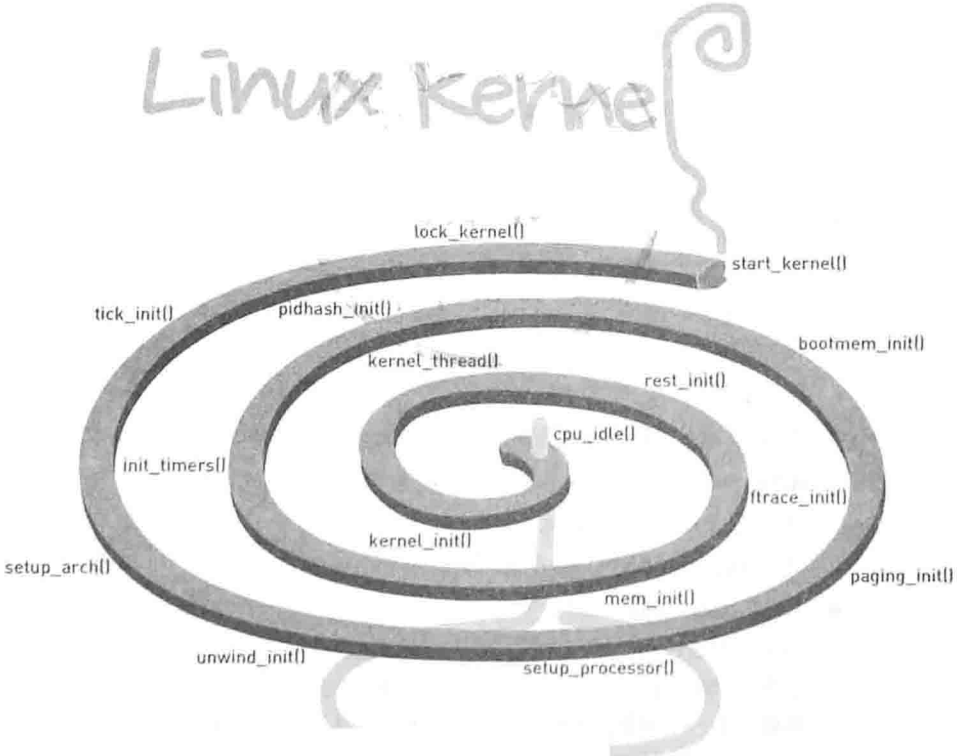
2年代码分析+1年执笔=第一本ARM Linux内核分析集大成之作！

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

ARM Linux 内核源码剖析

【韩】尹锡训等 著　崔范松 译



人民邮电出版社
北京

图书在版编目（CIP）数据

ARM Linux内核源码剖析 /（韩）尹锡训等著 ；崔范松译. -- 北京 ：人民邮电出版社，2014.7
（图灵程序设计丛书）
ISBN 978-7-115-35910-0

Ⅰ. ①A… Ⅱ. ①尹… ②崔… Ⅲ. ①微处理器－系统设计②Linux操作系统－系统设计 Ⅳ. ①TP332 ②TP316.89

中国版本图书馆CIP数据核字(2014)第117494号

内 容 提 要

本书是多位作者在3年Liunx内核分析经验和庞大资料基础上写成的，收录了其他同类书未曾讲解的内容并进行逐行分析，一扫当前市场中其他理论书带给读者的郁闷。书中详细的代码分析与大量插图能够使读者对Linux内核及ARM获得正确认识，自然而然习得如何有效分析定期发布的Linux内核。

本书适合想从Linux内核启动开始透彻分析全部启动过程的读者，因Linux代码量庞大而束手无策的人、想要了解Linux实际运行过程的人、渴求OS实操理论的人，本书必将成为他们不可或缺的参考书。

◆ 著 [韩] 尹锡训
译 崔范松
责任编辑 傅志红
执行编辑 陈 曦
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：33.25 插页：2
字数：786千字 2014年7月第1版
印数：1－4 000册 2014年7月北京第1次印刷
著作权合同登记号 图字：01-2013-8469号

定价：99.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

审阅者评语

整整3年的写作……

事实上，这本书的正式编写时间大约是1年。但我说用了3年，原因在于编写团队整整用了2年时间分析Linux代码，在此基础上，用1年完成了整个写作过程。为了这本书，他们在3年内不分昼夜地专注于Linux内核这一庞大领域，细心钻研，提高实力，并使辛劳努力的成果最终面世！

正是由于了解这些情况，所以，我收到原稿进行审阅时，比作者们更感慨万千。看着这本凝聚着编写团队辛勤汗水的书，我感到无比自豪。

简言之，这本书既是属于广大读者的书，更是作者们自己的书。因为书中是他们用2年时间分析Linux内核并不断提高实力的亲身经历，所以比现有的其他书籍水平更高。而且，编写团队已经完全掌握了书中内容，应该可以解答读者们的大部分疑惑。

这本书如同一把锋利的手术刀，大家读了就能感到，书中的每个部分都充满了鲜活的细节。这恰恰反映出2年间诸位作者一直保持着自己的实力。另外，由实力雄厚的编写团队逐一解开该主题的所有谜团，这使该书通顺流畅。

鉴于以上这些原因，我认为这本佳作比市面上出现的任何一本内核书都更出色。

我为这3年来付出最大努力的作者们喝彩！同时也为能够阅读此书进行学习的读者们感到幸运。因为我很久以前接触过Linux内核，但当时还没有这本书作参考，所以经历了无数失误。我向希望学习Linux内核的各位读者强烈推荐这本辅导书。

白昌佑

曾在三星电子、三星SDS、(株) Nusco公司主持开发各类RTOS，在三星综合技术院开发编译器，在(株) Nusco开发调试器和管理程序及各种系统S/W。目前在(株) Nusco担任CEO、软件开发团队指导。10年来一直负责管理离线系统S/W学习小组http://www.iamroot.org。

前　言

分析 Linux 内核并将其编撰成册，这对我们来说是一项巨大的挑战。通过 iamroot.org，我们在 2009 年 5 月举办了第一次以分析 ARM Linux 内核为目的的聚会。而且，完成第一次分析前的 2 年时间里，我们每周都是单休。身为工程师，能够亲自分析以前在书上接触过的操作系统代码，这让我们感到无比喜悦，可实际的分析过程却更要求我们耐心。

我们通过数年来对内核源代码分析的学习，依据整理好的 iamroot 的指导进行源代码分析。这本书主要讲述 ARM 使用的 Linux 内核源代码，逐行分析源代码和内核的实际工作内容，主要面向希望深层次了解内核内部机制的广大读者。逐行分析内核源代码主要是分析接通电源后如何调用内核的起始代码，以及调用起始代码内各种函数并最终显示命令行提示符的一系列过程。从这一点上讲，本书结构与其他 Linux 内核教程有所不同，涵盖了其他教材上没有涉及的内容。Linux 内核的基本概念可通过许多资料获得，但不操作内核源代码，就无法满足工程师们了解操作系统实现方法的求知欲。这正是本书最大的价值所在。

在分析初期，甚至到第一次分析结束时，写书这件事都不在我们的计划之中。大家开始分析内核时还兴致勃勃，逐步深入后才明白，我们的能力和知识是有限的，而编书这一重任更是远远超出了自己的能力范围。我们明知自身还有很多不足之处却依然坚持编写此书，就是希望将分析过程中遇到的困惑和辛辛苦苦理解到的内容与广大喜欢 Linux 的工程师、对内核感兴趣的读者分享。因为我们相信，正是有了许多工程师对 Linux 做出的贡献，才能使大家学习 Linux 内核代码。

我们通过本书与大家分享经验并不是因为自己掌握的知识多，也不因为这本书涉及的内容完美无缺。就像腼腆的林纳斯（Linus）1991 年用邮件与世界分享自己的内核代码那样，我们也只希望自己整理的内容能够给一些人带来帮助，也希望另一些人以这本书为基础，写出更多、更好的著作。所以，其实是我们在大家的帮助下更加深入地理解 Linux 内核。希望这本书能为想要了解 Linux 内核的广大读者带来些许帮助。

全体作者

2012 年 8 月

目　录

第一部分　ARM Linux 内核——分析内核前需要做的准备

第二部分　内核的启动——start_kernel 调用方法

第三部分 内核的执行——内核的起始与结束位置

附 录

第一部分

ARM Linux内核

——分析内核前需要做的准备

Linux 内核源代码即使压缩后也超过了 70 MB。对如此庞大的源代码进行逐行分析往往需要付出莫大的努力。但是，需要倾注的努力与我们具备的 Linux 知识、对 ARM 处理器的理解、代码分析工具的使用经验等或许是一种反比关系。也就是说，如果我们具备了这些丰富的知识和经验，将会大大缩短分析代码的时间。因此，正式进入源代码分析之前，需要做好充分准备。

第一部分是内核代码分析的准备阶段，主要包括对 Linux 内核的介绍、如何编译内核并生成可加载到内存的图像、必须掌握的 ARM 处理器相关内容、为提高代码分析效率而搭建分析环境等代码分析必备常识。

真正进入代码分析之前，我们利用第一部分做个热身吧！

第1章

内核介绍及2.6版和3.2版之间的差异

本书对启动Linux内核过程中调用的代码进行分析，并着重讲解Linux是怎样运行的，以及Linux运行时哪些部分需要初始化等内容。

本章在分析之前，先讲解两个内容：第一，Linux是由林纳斯发起的当前最成功的操作系统之一；第二，Linux内核是由多个子系统构成的一个单内核。

最后，了解本书中分析的内核版本和内核3.2版之间的差异。在这一过程中大家会发现，内核从2.4版升级到2.6版的过程中，结构上并没有发生太大变化。所以本书并没有把重点放在内核3.2版上，但对其的讲解有助于分析最新版的内核。

1.1 内核的诞生、作用以及内部结构

1.1.1 Linus 创造的 Linux

Linux[①]是由赫尔辛基大学的研究生林纳斯·托瓦兹（Linus Torvalds）在1991年发布的操作系统。林纳斯在大学期间对MINIX的许可授权政策感到不满，从而打算开发Linux。虽然开发初期是以MINIX为基础的，但进展到一定程度后，他在运行Linux内核的Linux系统上进行了开发。[②]

之后，与MINIX相关的组件全部被GNU应用程序取代[③]。由于从GNU阵营中得到了许多支持和无数内核黑客的贡献，Linux成为全世界应用最广、最为成功的开放源操作系统。

---

① Linux的早期名称是不一样的。刚开始林纳斯将自己制作的组件命名为Freax，后来接受了上传其内核的网站管理员的提议，更名为Linux，并使用至今。Linux的意思是“林纳斯的UNIX”（Linus's UNIX）或“Linux不是UNIX”（Linux is not UNIX），或者是“林纳斯的MINIX”（Linus's MINIX）。

② 构建Linux的编译器是linux-gcc。那么linux-gcc又是用什么编译器构建的呢？某位内核黑客与林纳斯对此进行过对话，参见如下URL：http://groups.google.com/group/comp.os.linux/msg/4ae6db18d3f49b0e。

③ 通过GNU获得了什么灵感或感动呢？在几乎与此同时的1992年年初，林纳斯用GPL取代了Linux的许可。

1.1.2 由多种子系统集成运行的单内核

Linux内核由许多子系统构成。图1-1表示各子系统之间的关系，请通过该图了解各子系统的作用。

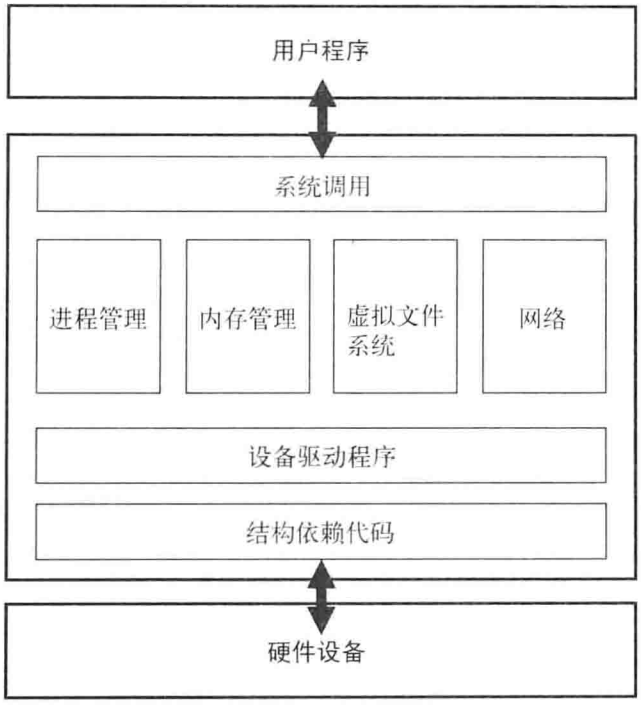


图1-1 Linux子系统结构

结构依赖代码

无论为何种结构类型，Linux均提供相同服务。但在其底层，每个结构都有需要另行控制的部分。这些部分是指用于CPU、MMU以及机载（on board）状态的低级（low-level）驱动程序，而且这些代码均位于arch目录下。

设备驱动程序

我们使用的电脑中，有很多周边设备（LCD、按钮、蓝牙、Wi-Fi等）通过多种方式（J2C、SCSI、EIDE）进行通信。设备驱动程序是指为这些高层设备制订的代码，位于drivers目录下。Linux内核中有一半以上是设备驱动程序代码，Linux通常最先封装最新设备。

进程管理

执行进程前必须先分配CPU资源。同时执行多个进程时，根据制定的政策公平分配资源，并且要管理进程的生成及状态转移的过程。进程管理就是执行这些任务的子系统，位于kernel目录下。

内存管理

与CPU同样重要的系统资源是内存。内存管理子系统负责内存分配、释放及共享，其相关代