国外经典教材·计算机科学与技术

IA-64 Linux® kernel design and implementation

IA-64 Linux® 内核设计与实现

David Mosberger
Stephane Eranian 著
梁金昆 等 译

清华大学出版社

IA-64 Linux® 内核设计与实现

David Mosberger
Stephane Eranian 著

梁金昆 等 译

清 华 大 学 出 版 社

北 京

EISBN: 0130610143

图书在版编目(CIP)数据

IA-64 Linux 内核设计与实现/摩斯博格(Mosberger, D.)，艾拉尼安(Eranian, S.)著；梁金昆等译. —北京：清华大学出版社，2004.11
书名原文：IA-64 Linux Kernel Design and Implementation
ISBN 7-302-09610-4

I. I… II. ①摩… ②艾… ③梁… III. Linux 操作系统 IV. TP316.89

中国版本图书馆 CIP 数据核字(2004)第 097098 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社总机：010-62770175　　**客户服务**：010-62776969
责任编辑：常晓波
封面设计：立日新
印 刷 者：清华园胶印厂
装 订 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：27.25　**字数**：671 千字
版　　次：2004 年 11 月第 1 版　2004 年 11 月第 1 次印刷
书　　号：ISBN 7-302-09610-4/TP·6665
印　　数：1～3000
定　　价：45.00 元

出 版 说 明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整和改革，急需一批门类齐全、具有国际先进水平的计算机经典教材，以适应当前我国计算机科学的教学需要。通过使用国外先进的经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培育出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构，Pearson 在高等教育领域有着不凡的表现，其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系，这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为该套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外经典教材·计算机科学与技术

编审委员会

前　言

Bruce Perens
Linux 与 Open Source 高级顾问
惠普(Hewlett-Packard)公司

本书是操作系统软件发展史上的一个里程碑，将介绍一种强大而又完全开放的操作系统，并采用独特视角分析了将其内核移植到一个崭新的 CPU 架构上的内核设计师们的设计思路。本书适用于学习操作系统编程的学生，也能教给经验丰富的内核程序员一两个窍门。本书从技术角度深入地介绍了现代 CPU 及其指令集和架构，以及 Linux 内核；并展示了现代微处理器的设计师们如何从效率和可伸缩性的角度来构建微处理器；另外还介绍了在与硬件无关的层次上内核的设计目标，以及介于与硬件无关的部分和 IA-64 架构之间的特定实现。本书涵盖了内核设计师在把 Linux 移植到 IA-64 的过程中所做出的全部决策，这展示了他们如何将硬件和软件整合为一个能正常运转的系统。

本书的作者之一跻身于能接替 Linus Torvalds 领导 Linux 内核开发的最后候选人之列。两位作者均受聘于 HP 的研究实验室，HP 公司创造了 IA-64 的直系祖先及其架构，后来与 Intel 合作开发 IA-64 架构。两位作者领导了 Linux 内核向 IA-64 处理器的移植，因此他们是讲述这一主题的最佳人选。

如果想从系统程序员的角度理解 IA-64 架构，或者想更深刻地理解 Linux 内核，或是面临着以下任务：改进软件对处理器的利用，把某个操作系统移植到一种新型的处理器上，乃至设计一种新型的 CPU，本书都将很适合你。本书为学习操作系统的学生提供了理论联系实际的重要桥梁。这面临着一个严峻的考验：保持架构整洁和可移植的抽象目标面临着必须在实际的 CPU 上高效且可靠运行的挑战。

但最具革命意义的一点是对于所有读者来说，本书实际上是完全可用的。仅仅几年前，本书的内容和相关源代码很可能还属于商业秘密，而拥有这些资料的公司估计它们具有数千万美元的价值。就算只是想看看 IA-64 内核源代码，就必须先在 HP 的少数几个部门之中找到一份工作。只是好奇的人或者学生是没有这样的机会的。实际上，在 20 世纪 90 年代初，AT&T 估计其 UNIX 系统的知识产权价值为 2.5 亿美元。资金雄厚的大学有时候会购买允许研究生使用 UNIX 源代码的授权，但是一旦如此，那些研究员便加入了一份令人反感的非公开协定。时代已经改变了：现在你已被授权使用该操作系统内核的全部源代码，甚至可以随意把它复制给你的朋友！

是什么使这一改变成为可能呢？原因就是 Linus Torvalds 和数百名合作者以源代码的形式公开了 Linux 内核，而最重要的是有了革命性的 GNU 通用公共许可(GNU GPL)。GPL

的自由软件(Free Software)模型允许对该系统及其源代码的自由使用和分发，允许任何人参加该系统的开发，使独立的开发人员和商业界结成伙伴，但同时限制了合作的任何一方不公平地利用另一方。欢迎你加入其中。

要理解 GPL 对 Linux 的重要性，必须要考虑 20 世纪 80 年代末到 90 年代初，UNIX 所面临的走向衰落的困境。商业人士深信 Microsoft NT 很快将一统天下。在那个时期，甚至连 Apple 和 NeXT 的创立者 Steve Jobs 都让步了，在其 Pixar 的台式机上安装了 Windows 系统，而不是他自己的产品。但下面两个因素拯救了 UNIX：Microsoft 承诺发布企业级 NT，但该计划延误了 5 年多的时间；而 GNU/Linux 系统复兴了 UNIX 的革命。

GNU 和 Linux 重新点燃了 UNIX 世界的希望，这完全是因为它们具有开放、共享的特性，并将其统一为一个能在所有厂商的硬件平台上运行的通用操作系统。UNIX 曾饱受过度分化的痛苦：为了紧紧抓住客户，每个厂商都对系统进行了修改，而厂商非常渴望挣回数量相当大的研发费用，以至于他们按用户数量对 UNIX 客户收费，并限制其系统只允许客户授权已付费的用户登录。而遵循 GPL 的 Linux 则不易受到强制分化和知识产权保护问题的困扰：GPL 的条款要求对软件的改进必须与所有人共享，而且规定拥有软件的任何人都能对其进行修改。一旦能修改软件，你就能去除对系统的任何限制。

GPL 对于共享和类似于共享的规定趋向于涵盖操作系统和底层硬件。这侵犯了厂商的利润，但完全符合客户的利益。实际效果是厂商开始把操作系统看作支持软件，而不是利润的直接来源。做到这一点是可能的，因为自由软件的协作特性意味着任何个人或公司都不需要过多地负担开发和服务的费用。公司可以与其直接的竞争对手共享这种无分化软件的开发，有些公司已经这样做了。例如，IBM 和 HP 虽然为 Linux 的市场份额而竞争，但它们都热衷于在许多自由软件项目上进行合作。通过这种方式，这两家公司可以共同完成无法独立靠各自的预算来进行的工作。

这就是自由软件革命的合理结果：你现在可以研究和修改那些几年前还是私有的成果，取得这些成果的人也不再保护其技术只限于其雇员使用，而会毫无保留地告诉你他们是如何取得这一成果的。本书从技术上深入地介绍了 IA-64 架构及其相应版本的 Linux 内核。因为你可以随心所欲地探索 Linux 系统，所以本书对你没有丝毫的隐瞒。通过这些课程，你能学会把 Linux 内核移植到另一种处理器上，向内核添加新的特性，乃至为 IA-64 编写一个全新的操作系统。你也可以选择只是更加深入地了解应用程序如何获得操作系统提供的服务，以及它们为何采取现有的运行方式。如何使用这些信息完全取决于你的选择。它们不再是实验室里的秘密，相关的知识产权保护问题也已不复存在，操作系统的大门被猛然地推开。革命已经到来：进来探索其中的奥秘吧。

——Bruce Perens

序　言

编写本书的目的非常简单，就是想精确地描述 Linux 在 IA-64 计算机上是如何工作的。通过实现这一目的，我们希望不仅能够突出 Linux 的内部工作方式，还能与大家分享一些令人兴奋的创造性的过程，其中涉及到解决为像 IA-64 这样的一个革命性的新平台设计操作系统时所遇到的诸多技术难题。IA-64 的许多革新的目的是通过赋予编译器更大的 CPU 控制权来提高性能。然而，我们将在本书中看到，IA-64 还提供了一种强大的系统架构，它支持(实际上是鼓励)操作系统级的创新的解决方案。

Linux 是一个实用的操作系统，始终遵循“实践检验真理”的原则。同时，Linux 总能快速地适应硬件和操作系统技术的实际发展。因此，尽管我们相信本书中描述的设计和实现提供了一个坚实的基础，但我们同样相信还存在着大量值得改进的地方。从这种意义上来说，我们鼓励围绕该平台所展开的研发工作。当然，Linux 是开放源码的操作系统这一事实，也有助于使其成为探索新想法的一个理想的试验平台。与那些推测的操作系统不同，Linux 提供了把灵机一动得到的解决方案快速付诸实践的机会。

现在我们知道了可以免费获得 Linux 的源代码，有人可能会问：直接通过阅读源代码来学习 Linux 及其针对 IA-64 的实现是不是不可能？源代码确实包含了对 Linux 工作原理的最精确的描述，但太精确有时也会带来坏处：这会让人只见树木而不见森林。另外，源代码能解释的只是如何实现，并不能解释为何以某种特定的方式来实现。而本书正好能弥补这些不足。

当引入一个新主题时，我们先会在较高的层次上介绍 Linux 所采用的方法的思想、原理和动机。接着会介绍 Linux 使用的硬件抽象接口，最后讲述其针对 IA-64 的实现。在这些讨论中，焦点始终是需要实现什么。我们也会介绍如何去实现，但会在较高的层次上，这样就能避免使读者陷于细节之中。这就意味着使用本书的方式主要有两种：对于想了解 Linux/ia64 概况的读者，这是一份完备而又权威的描述；对于想亲手实践的读者，它可以使你顺利地进入 Linux/ia64 及其底层源代码的世界。为了帮助想亲手实践的读者，本书中的文字描述包含指向 Linux 源代码的关键部分的指示。这些指示对于版本为 v2.4.14 的 Linux 内核来说是最准确的，但也适用于早期的和稍后的版本。

面向的读者

本书主要面向那些想深入了解 Linux 内核在 IA-64 上如何工作的专业人士。另外，对于硬件设计师，本书作为特定操作系统如何利用 IA-64 的一个案例分析，也是非常有用的。同样，对于想了解如何为 IA-64 设计其他操作系统的软件设计师，本书提供了许多提示，这些提示是关于如何处理 IA-64 的某些更为先进的特性。例如，本书讨论了猜测执行、寄存器堆栈和虚拟哈希页表 walker 对操作系统的影响。最后，本书对真实的通用操作系统如何在真实的硬件上工作的描述，也能吸引计算机科学与工程专业的学生。

对于 Linux 专业人士，本书最适合那些希望调整 IA-64 性能的普通的内核程序员、设备驱动程序的开发人员以及应用程序的开发人员。除此之外，对 Linux 硬件抽象接口的描述与目标平台无关。我们相信本书中对接口的描述是现有的最精确且最全面的描述之一。但也应该注意到这些接口是由一组人数众多的开发人员通过长期的工作而开发出来的，在某种程度上讲该工作还在继续。因此，尽管我们尽可能地使本书的描述涵盖的范围更广一些，但不能宣称它们对于 IA-64 以外的平台来说是权威的。

关于 C 程序设计语言方面的知识是阅读本书的先决条件。熟悉基本的操作系统概念、汇编程序设计和计算机组成原理也是有帮助的，但对这些方面并不苛求。阅读本书不需要关于 IA-64 的预备知识。

章节组织

本书的前两章是介绍性的资料。第 1 章引言，介绍与微处理器架构和 Linux 这两者的发展相关的背景知识。第 1 章的后半部分是对 Linux 内核的概述。作为第 1 章概述的一部分，我们还给出了本书常用的术语。

第 2 章 IA-64 架构，介绍 IA-64 的架构和软件约定。IA-64 是一个有丰富内涵的架构，对它的学习不可能一蹴而就。因此，我们建议开始时快速浏览第 2 章，而不要过于关注细节。一旦对该架构基本熟悉之后，读者可能就会时常需要重读第 2 章并更详细地研究某些特定的问题。

接下来的 3 章介绍 Linux 最基本的组件以及它们在 IA-64 上如何工作：第 3 章进程、任务和线程，介绍关于调度和执行方面的问题。第 3 章首先概述关键的数据结构，然后描述 Linux 线程接口和 Linux 所支持的各种同步原语。第 4 章虚拟内存，介绍 Linux 的虚拟内存系统。第 4 章在简要介绍之后详细地描述了所有的硬件抽象接口及其针对 IA-64 的实现。第 4 章涵盖的主题有：Linux 页表、线性映射的虚拟页表、TLB(旁路转换缓冲)管理、页面错误处理，以及内存一致性问题。第 5 章内核入口与出口，介绍与进入和退出内核相关的各个方面。特别需要指出的是，第 5 章会解释系统调用和信号如何工作，以及数据是如何在用户/内核范围内进行传递的。如同前两章一样，第 5 章首先介绍 Linux 适用于所有平台的各个方面，然后以描述针对 IA-64 的具体实现来完成对相关问题的讨论。

接下来的 3 章彼此之间相对独立。第 6 章栈展开，讨论堆栈展开这个一般性的主题。尽管实际上该主题并不是 IA-64 所特有的，但它在该平台上起着更重要的作用。确实，所有想编写IA-64汇编代码的人都必须熟悉第6章第3节的内容。第6章的其他小节介绍IA-64的内核展开程序(unwinder)及其实现。第 7 章 I/O 设备，描述关于设备输入/输出(I/O)方面的问题。第 7 章还特别包括了用于编程 I/O 的(内存映射式和基于端口的)硬件抽象接口、DMA(直接内存访问)，以及设备中断。我们先描述接口，然后介绍其针对 IA-64 的实现。第 8 章对称多处理，讨论关于多处理器(MP)计算机的特定方面的问题。虽然其他章节在一般性的讨论过程中于适当的地方涉及了 MP 方面的问题，但第 8 章涵盖了其余的所有问题。特别需要指出的是，第 8 章第 1 节概述了 Linux 的加锁原理和 MP 支持接口。其余的两节介绍对特定于 CPU 的数据区域的处理和在 MP 计算机上高分辨率时间戳的维护问题。

最后 3 章的内容主要特定于 IA-64。第 9 章系统性能，介绍 IA-64 对性能监控的支持及

其针对 Itanium(安腾处理器)的实现，和相关的 perfmon 内核子系统。对于希望刻画和调整 IA-64 程序(无论是普通应用程序还是 Linux 内核的某些部分)性能的人，第 9 章具有很高的参考价值。第 10 章启动，介绍与启动计算机相关的各个方面。前两节概述了 IA-64 的固件(firmware)和引导装入程序(bootloader)。第三节介绍 Linux 的自举接口及其针对 IA-64 的实现。第 11 章 IA-32 兼容性，描述 Linux/ia64 如何设法提供对 IA-32 的向后兼容。这些讨论当然是特定于 IA-64 的，然而，大多数 64 位的 Linux 平台提供了对某些 32 位平台的向后兼容。因此，这里讨论的很多主题和解决方案也适用于其他平台。

本书中使用的术语和缩写在附录 E 的术语表中给出。当遇到不熟悉的缩写时，建议读者参考该术语表。

致　谢

许多人直接或间接地对本书做出了贡献。

首先，要感谢 HP 的 Gary Coutant，Dale Morris，Dong Wei，Jerry Huck，Jim Callister 和 Jim Hull 以及 Intel 的 Asit K. Mallick 和 Rumi Zahir，他们耐心地解答了我们提出的关于 IA-64 和 Itanium（安腾）CPU 的一些较深奥的问题。

HP 的 Brian Lynn、Hans Boehm 和 Khalid Aziz 负责正式审阅本书，非常感谢他们提供的反馈。尤其是 Brian，花费了大量的时间来审阅每章的多次修订，每次总能提出具有深刻见解的意见。我们很幸运地得到了几位志愿审阅者的帮助。下面以所审阅的章节的顺序一一列出我们要感谢的人：CERN 的 Sverre Jarp 审阅了关于 modulo-scheduled 循环硬件的内容，Cary Coutant 对 IA-64 软件约定的相关叙述提出了有益的修改意见。Urbana-Champaign 大学的 David K. Raila 和 HP 的 Christophe de Dinechin 审阅了虚拟内存这一章。Cygnus Solutions（现在是 Red Hat Solutions）的 Kevin Buettner 审阅了栈展开这一章。Intel 的 Asit K. Mallick 审阅了设备 I/O 和多处理这两章，澄清了一些概念并提供了有益的见解。Wild Open Source 的 Jes Sørensen 也审阅了设备 I/O 这一章，并帮助我们找出了本章书稿中的一个错误。HP 的 Jim Callister 不但帮助审阅了系统性能这一章的内容，还帮助我们揭开了 Itanium 性能监控单元背后的一些秘密。要特别感谢 HP 的 Dong Wei 审阅了启动这一章中的 ACPI 那一节。感谢 Independent Storage 的 Don Dugger 审阅了 IA-32 兼容性这一章。Intel 的 Sunil Saxena 为我们解答了 IA-32 的 ELF 二进制为何使用如此奇怪的加载地址这一难题。

非常感激所有的审阅者，感谢他们做出的巨大贡献。同时，需要强调的一点是，对于本书的准确性我们责无旁贷。如果读者发现任何错误，请不吝赐教，我们不胜感激。我们的联系方式在“提供反馈部分”。

对于本书的发行，要感谢 HP Books Publishing 的 Susan Wright 和 Pat Pekary。Susan 帮助本书走上正轨。还要感谢 Prentice Hall PTR 的组稿编辑 Jill Pisoni Harry 和发行编辑 Jane Bonnell。他们与足智多谋的同事们一起帮助我们出版了这本书，使我们无须分心于此。

最后，要衷心感谢 Linus Torvalds 创造了 Linux，感谢 Linux 团体造就了这样一个了不起的开放源码操作系统。出于同样的原因，要感谢 HP Labs 的管理层给予我们编写这本书的自由和所需的充足的时间。

本书的原稿和版面是使用在 Linux 上运行的 LaTeX 2ε，xfig，GNU Emacs 和 X Window 系统编辑而成。本书的大部分内容是在 HP 的 OmniBook 5700CTX 笔记本电脑上编写的，而最终的版面在一台 HP i2000 Itanium 工作站上完成。特别要感谢 xfig 软件包的维护者 Brian V. Smith，他调整了 xfig，从而简化了本书的出版工作。

提供反馈

我们尽可能地提高本书的趣味性、使用价值和准确性，同时也希望读者能针对如何改进将来的版本提出反馈意见。为此，我们建立了一个 Web 站点，地址（URL）如下：

http://www.lia64.org/book/

欢迎访问该站点，报告错误或提出改进意见。该 Web 站点还提供了最新的勘误表、相关软件和工具的链接，以及 Linux 的发展概要(这部分内容包含在本书中)。

David Mosberger，Stephane Eranian
2001 年 11 月于加利福尼亚 Palo Alto

目　录