

序言

本书是一本有关 Linux 操作系统内核基本工作原理的入门读物。

本书的主要目标

本书的主要目标是使用尽量少的篇幅或在有限的篇幅内，对完整的 Linux 内核源代码进行解剖，以期对操作系统的基本功能和实际实现方式获得全方位的理解。做到对 linux 内核有一个完整而深刻的理解，对 linux 操作系统的基本工作原理真正理解和入门。

本书读者群的定位是一些知晓 Linux 系统的一般使用方法或具有一定的编程基础，但比较缺乏阅读目前最新内核源代码的基础知识，又急切希望能够进一步理解 UNIX 类操作系统内核工作原理和实际代码实现的爱好者。这部分读者的水平应该介于初级与中级水平之间。目前，这部分读者人数在 Linux 爱好者中所占的比例是很高的，而面向这部分读者以比较易懂和有效的手段讲解内核的书籍资料不多。

现有书籍不足之处

目前已有的描述 Linux 内核的书籍，均尽量选用最新 Linux 内核版本（例如 Redhat 7.0 使用的 2.2.16 稳定版等）进行描述，但由于目前 Linux 内核整个源代码的大小已经非常得大（例如 2.2.20 版具有 268 万行代码！），因此这些书籍仅能对 Linux 内核源代码进行选择性地或原理性地说明，许多系统实现细节被忽略。因此并不能给予读者对实际 Linux 内核有清晰而完整的理解。

Scott Maxwell 著的一书《Linux 内核源代码分析》（陆丽娜等译）基本上是对 Linux 中级水平的读者，需要较为全面的基础知识才能完全理解。而且可能是由于篇幅所限，该书并没有对所有 Linux 内核代码进行注释，略去了很多内核实现细节，例如其中内核中使用的各个头文件(*.h)、生成内核代码映像文件的工具程序、各个 make 文件的作用和实现等均没有涉及。因此对于处于初中级水平之间的读者来说阅读该书有些困难。

浙江大学出版的《Linux 内核源代码情景分析》一书，也基本有这些不足之处。甚至对于一些具有较高 Linux 系统应用水平的计算机本科高年级学生，由于该书篇幅问题以及仅仅选择性地讲解内核源代码，也不能真正吃透内核的实际实现方式，因而往往刚开始阅读就放弃了。这在本人教学的学生中基本都会出现这个问题。该书刚面市时，本人曾极力劝说学生购之阅读，并在二个月后调查阅读学习情况，基本都存在看不下去或不能理解等问题，大多数人都放弃了。

John Lions 著的《莱昂氏 UNIX 源代码分析》一书虽然是一本学习 UNIX 类操作系统内核源代码很好的书籍，但是由于其采用的是 UNIX V6 版，其中系统调用等部分代码是用早已过时的 PDP-11 系列机的汇编语言编制的，因此在阅读与硬件部分相关的源代码时就会遇到较大的困难。

A.S.Tanenbaum 的书《操作系统：设计与实现》是一本有关操作系统内核实现很好的入门书籍，但该书所叙述的 MINIX 系统是一种基于消息传递的内核实现机制，与 Linux 内核的实现有所区别。因此在学习该书之后，并不能很顺利地即刻着手进一步学习较新的 Linux 内核源代码实现。

在使用这些书籍进行学习时会有一种“盲人摸象”的感觉，不能真正理解 Linux 内核系统具体实现的整体概念，尤其是对那些 Linux 系统初学者或刚学会如何使用 Linux 系统的人在使用那些书学习内核

原理时，内核的整体运作结构并不能清晰地脑海中形成。这在本人多年的 Linux 内核学习过程中也深有体会。在 1991 年 10 月份，Linux 的创始人 Linus Torvalds 在开发出 Linux 0.03 版后写的一篇文章中也提到了同样的问题。在这篇题为“**LINUX--a free unix-386 kernel**”的文章中，他说：“开发 Linux 是为了那些操作系统爱好者和计算机科学系的学生使用、学习和娱乐。...自由软件基金会的 GNU Hurd 系统如果开发出来就已经显得太庞大而不适合学习和理解。”而现今的 Linux 系统要比 GNU 的 Hurd 系统更为庞大和复杂，因此同样也已经不适合作为操作系统初学者的入门学习起点。这也是写作本书的动机之一。

为了填补这个空缺，本书的主要目标是使用尽量少的篇幅或在有限的篇幅内，对完整的 Linux 内核源代码进行全面解剖，以期对操作系统的基本功能和实际实现方式获得全方位的理解。做到对 Linux 内核有一个完整而深刻的理解，对 Linux 操作系统的基本工作原理真正理解和入门。

阅读早期内核的其它好处

目前，已经出现不少基于 Linux 早期内核而开发的专门用于嵌入式系统的内核版本，如 DJJ 的 x86 操作系统、Uclinux 等（在 www.linux.org 上有专门目录），世界上也有许多人认识到通过早期 Linux 内核源代码学习的好处，目前国内也已经有人正在组织人力注释出版类似本文的书籍。因此，通过阅读 Linux 早期内核版本的源代码，的确是学习 Linux 系统的一种行之有效的途径，并且对研究和应用 Linux 嵌入式系统也有很大的帮助。

在对早期内核源代码的注释过程中，作者发现，早期内核源代码几乎就是目前所使用的较新内核的一个精简版本。其中已经包括了目前新版本中几乎所有的基本功能原理的内容。正如《系统软件：系统编程导论》一书的作者 Leland L. Beck 在介绍系统程序以及操作系统设计时，引入了一种极其简化的简单指令计算机(SIC)系统来说明所有系统程序的设计和实现原理，从而既避免了实际计算机系统的复杂性，又能透彻地说明问题。这里选择 Linux 的早期内核版本作为学习对象，其指导思想与 Leland 的一致。这对 Linux 内核学习的入门者来说，是最理想的选择之一。能够在尽可能短的时间内深入理解 Linux 内核的基本工作原理。

对于那些已经比较熟悉内核工作原理的人，为了能让自己在实际工作中对系统的实际运转机制不产生一种空中楼阁的感觉，因此也有必要阅读内核源代码。

当然，使用早期内核作为学习的对象也有不足之处。所选用的 Linux 早期内核版本不包含对虚拟文件系统 VFS 的支持、对网络系统的支持、仅支持 a.out 执行文件和对其它一些现有内核中复杂子系统的说明。但由于本书是作为 Linux 内核工作机制实现的入门教材，因此这也正是选择早期内核版本的优点之一。通过学习本书，可以为进一步学习这些高级内容打下坚实的基础。

阅读完整源代码的重要性和必要性

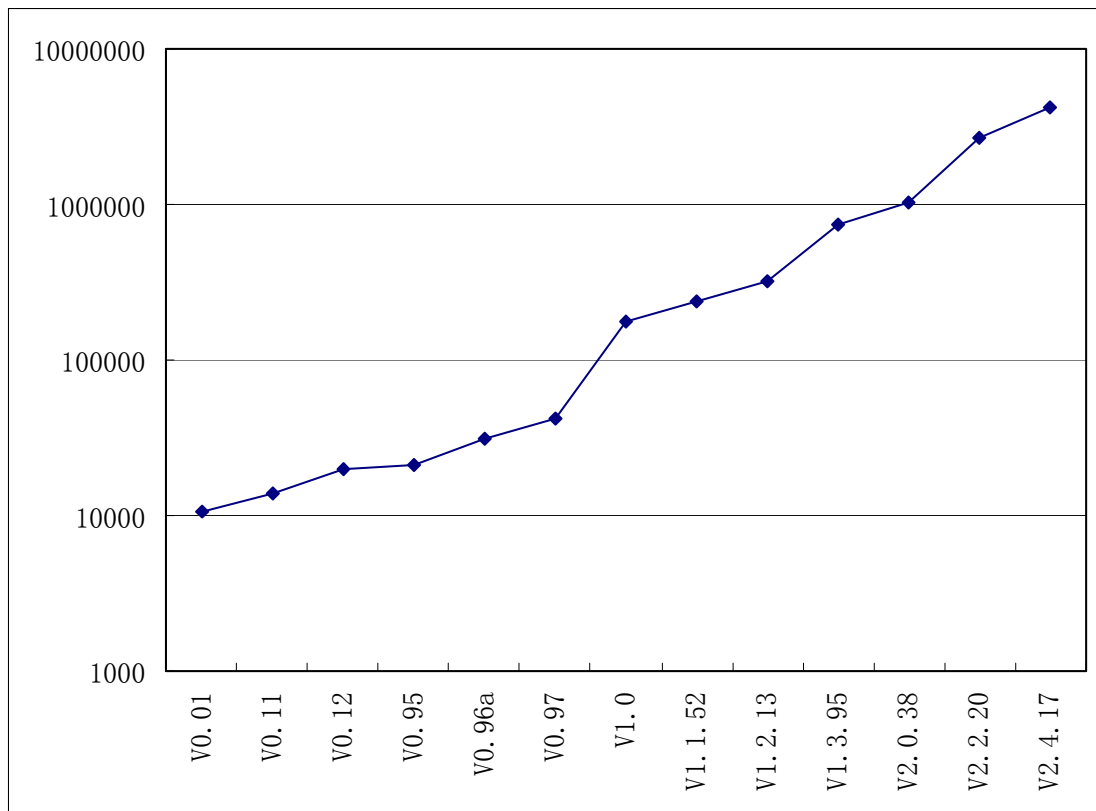
正如 Linux 系统的创始人的一篇新闻组投稿上所说的，要理解一个软件系统的真正运行机制，一定要阅读其源代码（RTFSC – Read The Fucking Source Code）。系统本身是一个完整的整体，具有很多看似不重要的细节存在，但是若忽略这些细节，就会对整个系统的理解带来困难，并且不能真正了解一个实际系统的实现方法和手段。

虽然通过阅读一些操作系统原理经典书籍（例如 M.J.Bach 的《UNIX 操作系统设计》）能够对 UNIX 类操作系统的工作原理有一些理论上的指导作用，但实际上对操作系统的真正组成和内部关系实现的理解仍不是很清晰。正如 AST 所说的，“许多操作系统教材都是重理论而轻实践”，“多数书籍和课程为调度算法耗费大量的时间和篇幅而完全忽略 I/O，其实，前者通常不足一页代码，而后者往往要占到整个系统三分之一的代码总量。”内核中大量的重要细节均未提到。因此并不能让读者理解一个真正的操作系统实现的奥妙所在。只有在详细阅读过完整的内核源代码之后，才会对系统有一种豁然开朗的感觉，对

整个系统的运作过程有深刻的理解。以后再选择最新的或较新内核源代码进行学习时，也不会碰到大问题，基本上都能顺利地理解新代码的内容。

如何选择要阅读的内核代码版本

那么，如何选择既能达到上述要求，又不被太多的内容而搞乱头脑，选择一个适合的 Linux 内核版本进行学习，提高学习的效率呢？作者通过对大量内核版本进行比较和选择后，最终选择了与目前 Linux 内核基本功能较为相近，又非常短小的 0.11 版内核作为入门学习的最佳版本。下图是对一些主要 Linux 内核版本行数的统计。



目前的 Linux 内核源代码量都在几百万行的数量上，极其庞大，对这些版本进行完全注释和说明几乎是不可能的，而 0.11 版内核不超过 2 万行代码量，因此完全可以在一本书中解释和注释清楚。麻雀虽小，五脏俱全。

另外，使用该版本可以避免使用现有较新内核版本中已经变得越来越复杂得各子系统部分的研究（如虚拟文件系统 VFS、ext2 或 ext3 文件系统、网络子系统、新的复杂的内存管理机制等）。

阅读本书需具备的基础知识

在阅读本书时，作者希望读者具有以下一些基础知识或有相关的参考书籍在手边。其一是有关 80x86 处理器结构和编程的知识或资料。例如可以从网上下载的 80x86 编程手册（INTEL 80386 Programmer's Reference Manual）；其二是有关 80x86 硬件体系结构和接口编程的知识或资料。有关这方面的资料很多；其三还应具备初级使用 Linux 系统的简单技能。

另外，由于 Linux 系统内核实现，最早是根据 M.J.Bach 的《UNIX 操作系统设计》一书的基本原理

开发的，源代码中许多变量或函数的名称都来自该书，因此在阅读本书时，若能适当参考该书，更易于对源代码的理解。

Linux 在最初开发 Linux 操作系统时，参照了 MINIX 操作系统。例如，最初的 Linux 内核版本完全照搬了 MINIX 1.0 文件系统。因此，在阅读本书时，A.S.Tanenbaum 的书《操作系统：设计与实现》也具有较大的参考价值。但 Tanenbaum 的书描述的是一种基于消息传递在内核各模块之间进行通信（信息交换），这与 Linux 内核的工作机制不一样。因此可以仅参考其中有关一般操作系统工作原理章节和文件系统实现的内容。

使用早期版本是否过时？

表面看来本书对 Linux 早期内核版本注释的内容犹如 Linux 操作系统刚公布时 Tanenbaum 就认为其已经过时的（Linux is obsolete）想法一样，但通过学习本书内容，你就会发现，利用本书学习 Linux 内核，由于内核源代码量短小而精干，因此会有极高的学习效率，能够做到事半功倍，快速入门。并且对继续进一步选择新内核部分源代码的学习打下坚实的基础。在学习完本书之后，你将对系统的运作原理有一个非常完整而实际的概念，这种完整概念能使人很容易地进一步选择和学习新内核源代码中的任何部分，而不需要再去啃读代码量巨大的新内核中完整的源代码。

Ext2 文件系统与 MINIX 文件系统

目前 Linux 系统上所使用的 Ext2（或最新的 Ext3）文件系统，是在内核 1.x 之后开发的文件系统，其功能详尽并且其性能也非常完整和稳固，是目前 Linux 操作系统上默认的标准文件系统。但是，作为对 Linux 操作系统完整工作原理入门学习所使用的部分，原则上是越精简越好。为了达到对一个操作系统有完整的理解，并且能不被其中各子系统复杂性和过多的细节喧宾夺主，在选择学习剖析用的内核版本时，只要系统的部分代码内容能说明实际工作原理，就越简单越好。

Linux 内核 0.11 版上当时仅包含最为简单的 MINIX 1.0 文件系统，对于理解一个操作系统中文件系统的实际组成和工作原理已经足够。这也是选择 Linux 早期内核版本进行学习的主要原因之一。

在完整阅读完本书之后，相信您定会发出这样的感叹：“对于 Linux 内核系统，我现在终于入门了！”。此时，您应该有十分的把握去进一步学习最新 Linux 内核中各部分的工作原理和过程了。

同济大学
赵炯 博士
2002.12