

中国开源软件系列丛书 ➤

Linux 内核分析

——基于Linux 0.01版本

工业和信息化部软件与集成电路促进中心 编著

- ◆ **案例典型** 选用最典型的Linux 0.01版本进行分析，降低学习难度的同时可让读者更容易体会到学习的乐趣。
- ◆ **深入浅出** 案例完整详细，代码清晰易读，语言通俗易懂，由浅入深，循序渐进。



中国水利水电出版社
www.waterpub.com.cn

中国开源软件系列丛书

Linux 内核分析——基于 Linux 0.01 版本

工业和信息化部软件与集成电路促进中心 编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书针对 Linux 的内核分析, 由浅入深地进行诠释, 内容丰富、循序渐进、通俗易懂、实用性强。本书重点是 Linux 内核分析, 让读者能够更清晰地理解 Linux 内核源码。书中精选的 Linux 0.01 内核是最简单、最精致的 Linux 内核, 可以帮助读者快速掌握 Linux 内核设计原理, 特别适合 Linux 的初学者阅读。

本书可作为 Linux 初学者的最佳入门教材, 也可供计算机爱好者和软件工程师学习、分析 Linux 内核使用。

图书在版编目 (C I P) 数据

Linux内核分析: 基于Linux 0.01版本 / 工业和信息化部软件与集成电路促进中心编著. -- 北京: 中国水利水电出版社, 2011.8 (2015.5重印)
(中国开源软件系列丛书)
ISBN 978-7-5084-8891-2

I. ①L… II. ①工… III. ①Linux操作系统 IV. ①TP316.89

中国版本图书馆CIP数据核字(2011)第167378号

策划编辑: 周春元 责任编辑: 张玉玲

书 名	中国开源软件系列丛书 Linux 内核分析——基于 Linux 0.01 版本
作 者	工业和信息化部软件与集成电路促进中心 编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16开本 8.25印张 210千字
版 次	2011年8月第1版 2015年5月第3次印刷
印 数	0001—2000册
定 价	28.00元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

中国开源软件系列丛书

编审委员会

特别顾问：

陆首群 中国开源软件推进联盟名誉主席
倪光南 中国工程院院士

主任委员：

邱善勤 工业和信息化部软件与集成电路促进中心主任

副主任委员：

宋茂强 北京邮电大学软件学院院长
韩乃平 中标软件有限公司总经理
刘 澎 中国开源软件推进联盟秘书长
高松涛 工业和信息化部软件与集成电路促进中心副主任

委 员：

刘龙庚 工业和信息化部软件与集成电路促进中心主任助理
陈 钟 北京大学软件与微电子学院教授
吴中海 北京大学软件与微电子学院副院长
高 林 工业和信息化部电子标准化研究所信息技术研究中心主任
宫 敏 Linux 内核专家
刘 明 工业和信息化部软件与集成电路促进中心软件处副处长
金友兵 北京红旗中文贰仟软件技术有限公司副总经理
郑忠源 北京中科红旗软件技术有限公司副总裁
陈 渝 清华大学计算机科学与技术系副教授
杨春燕 共创软件联盟常务副秘书长
王威信 工业和信息化部软件与集成电路促进中心战略研究部主任
杜 娇 工业和信息化部软件与集成电路促进中心知识产权部主任
唐仕武 工业和信息化部软件与集成电路促进中心软件处副处长
张 靖 北京赛普信科技有限公司教育培训部副主任
冯晓焰 英特尔亚太研发有限公司首席开源科学家
胡昌军 中国开源软件中心技术总监
陶 品 清华大学计算机科学与技术系副教授
张 俊 北京共创开源软件有限公司产品总监
张永军 中标软件有限公司产品经理
陈 绪 英特尔中国有限公司总监
杨 松 北京红旗中文贰仟软件技术有限公司产品部总监
李 恒 北京红旗中文贰仟软件技术有限公司技术总监

II

编委会

本书主编：刘龙庚

本书编委：

胡昌军 杨东日 王威信

孙加兴 杜 娇 王少华

吴 桐 王 莉 陈 越

随着 Linux 的不断发展与成熟，越来越多的用户使用 Linux 作为服务器操作系统和桌面操作系统。Linux 操作系统内核原理与技术，已经成为计算机软件开发人员、信息系统建设人员的必备知识和技能。

本书特点

本书基于 Linux 0.01 版本详细介绍操作系统的基本功能、设计的基本原则、操作系统引导的基本知识、进程的概念与调度、存储管理和设计、设备管理与调度、磁盘文件系统、shell 编程技术、系统调用等，针对各项内容列举了大量丰富实用的 Linux 源码实例，针对源码进行了详细分析。书中的每个知识点都是以简短的篇幅介绍其中最基本、最常用的内容。让读者深刻理解并掌握世界上最优秀的操作系统之一——Linux 的内核，为从事 Linux 程序设计、操作系统设计、软件开发等工作打下坚实的基础。

概括来讲，本书具有如下特点：

- 取材广泛、内容丰富
- 案例完整、结构清晰
- 讲解通俗、步骤详细
- 代码准确、注释清晰

另外，“提示”中所列的均为读者容易产生混淆的内容或是对实际操作有帮助的一些经验性的方法和技巧。

读者对象

- 高等院校相关专业学生
- 高等职业技术学院相关专业学生
- 计算机培训教师和学员
- 计算机爱好者和相关技术人员
- Linux 程序员

编者

2011年6月

IV

目录

前言

第1章 Linux 和 GNU 简介	1	3.3.4 init 目录	20
1.1 自由的天地: GNU 和 Linux	1	3.3.5 kernel 目录	21
1.1.1 热爱和享受自由	2	3.3.6 lib 目录	21
1.1.2 神奇的 Linux	3	3.3.7 mm 目录	22
1.2 在乐趣中获得成功: 一起来玩操作系统	5	3.3.8 tools 目录	22
1.2.1 十一岁开始编程序	6	3.4 Linux 0.01 的 main.c 分析	22
1.2.2 一种操作系统的诞生	6	3.5 编译和运行 Linux 0.01 系统	26
1.3 GNU 的开发工具	8	本章小结	27
本章小结	8	第4章 操作系统引导——装载内核	28
第2章 操作系统设计入门	9	4.1 操作系统引导基本知识	28
2.1 操作系统介绍	9	4.1.1 系统引导设备	28
2.2 操作系统的基本功能	10	4.1.2 启动过程简介	29
2.3 操作系统设计的基本准则	10	4.1.3 硬盘主引导扇区简介	30
2.3.1 简单就是优美	11	4.2 Linux 0.01 引导代码分析	33
2.3.2 利用已有的基础	11	4.2.1 boot.s: BIOS 引导代码	33
2.3.3 良好的设计	11	4.2.2 head.s: Linux 的 32 位引导代码	33
2.3.4 单内核操作系统和微内核操作系统	12	4.3 实例: 使用 nasm 生成引导代码	33
本章小结	12	4.3.1 引导代码基础	34
第3章 Linux 0.01 内核	13	4.3.2 最简单的引导程序	34
3.1 Linux 0.01 内核简介	13	4.3.3 如何设置段寄存器	36
3.1.1 Linux 0.01 内核背景	14	4.3.4 在引导程序中装入程序	
3.1.2 Linux 0.01 的技术特色	14	(基于扇区)	36
3.1.3 存储管理	15	本章小结	40
3.1.4 文件系统	15	第5章 进程管理和调度	41
3.1.5 硬件平台移植和应用程序	16	5.1 进程的概念	41
3.2 Linux 0.01 核心代码目录	16	5.1.1 进程的定义	41
3.3 核心源代码的目录分布	17	5.1.2 进程的属性	42
3.3.1 boot 目录	17	5.2 进程调度	43
3.3.2 fs 目录	18	5.2.1 进程调度算法的选择	43
3.3.3 include 目录	19	5.2.2 进程描述符	43

5.2.3	Linux 进程调度源代码分析	44	7.1.2	驱动程序	71
5.2.4	进程的切换	45	7.2	Linux 0.01 中断处理	73
5.3	Linux 0.01 内核中的进程调度实例	46	7.2.1	中断处理的基本过程	73
5.3.1	进程调度	46	7.2.2	traps.c 文件分析	74
5.3.2	用户进程和内核线程	47	7.2.3	/kernel/asm.s 文件分析	74
	本章小结	49	7.3	如何编写中断服务程序 ISR	74
第 6 章	存储管理和设计	50	7.4	设备驱动程序实例——利用 BIOS 的 探测系统设备	76
6.1	存储管理概述	50	7.5	Linux 0.01 设备驱动程序源代码分析	77
6.1.1	信息的二级存储	51	7.5.1	终端控制代码: console.c	77
6.1.2	存储管理的功能	51	7.5.2	rs323 驱动代码: serial.c 和 rs_io.s	77
6.2	段页式存储管理	52		本章小结	77
6.2.1	页式存储管理	52	第 8 章	磁盘文件系统	78
6.2.2	段式存储管理	53	8.1	硬盘驱动器结构简介	79
6.2.3	段页式存储管理	54	8.1.1	硬盘参数解释	79
6.2.4	3 种存储管理方式的对比	55	8.1.2	基本 Int 13H 调用简介	79
6.3	物理存储管理	55	8.1.3	扩展 Int 13H 简介	79
6.3.1	技术细节	56	8.2	UNIX 文件系统分析	80
6.3.2	物理存储管理器的组织	57	8.3	VFS 简介	81
6.3.3	物理存储管理器的初始化	59	8.3.1	VFS 的体系结构	82
6.3.4	页面分配	59	8.3.2	文件的表示	83
6.3.5	页面回收	60	8.3.3	磁盘布局	83
6.3.6	映射	60	8.4	文件系统设计步骤	84
6.3.7	内存映射	61	8.4.1	步骤一: 磁盘函数库	84
6.4	虚拟存储管理	62	8.4.2	步骤二: 文件系统	85
6.4.1	技术细节	63	8.4.3	步骤三: 容错性	87
6.4.2	malloc()和 free()	63	8.5	实践: 自己编写的简单文件系统	87
6.4.3	缺页处理	64	8.6	Linux 0.01 文件系统源代码分析	88
6.4.4	虚拟存储管理的页面分配	65		本章小结	88
6.4.5	可执行程序	66	第 9 章	shell 编程技术和实例	90
6.4.6	交换	66	9.1	shell 的基本概念	90
6.4.7	总结	67	9.2	例子: 最简单的 shell 程序	92
6.5	malloc()和 free()的实现	68	9.3	管道和 I/O 重定向	97
6.6	Linux 0.01 存储管理代码分析	68	9.3.1	使用 dup()重定向 I/O	97
6.6.1	memory.c 分析	68	9.3.2	使用 pipe()和 dup()	98
6.6.2	page.s 分析	68	9.3.3	使用 dup2()的例子	99
	本章小结	69	9.4	t-shell 实现代码分析	101
第 7 章	设备管理和调度	70	9.4.1	shell 总体结构 (不支持管道)	101
7.1	Linux 设备管理概述	70	9.4.2	shell 总体结构 (支持管道)	102
7.1.1	设备管理的基本要求	71			

9.4.3	main 函数	102	10.1.1	系统调用	114
9.4.4	initialization()函数	103	10.1.2	系统调用实现	115
9.4.5	getline()函数	104	10.1.3	系统调用到 int 0x80 中断 请求的转换	121
9.4.6	parse()函数	105	10.2	Linux 0.01 系统调用实现分析	122
9.4.7	command()函数	107	10.2.1	system_call.s	122
9.4.8	execute()函数	108	10.2.2	sys.c	122
9.4.9	forkexec()函数	110	10.3	实例：在 Linux 中添加新系统调用	122
9.4.10	check()函数	111	10.3.1	实现系统调用代码	122
9.4.11	getname()函数	112	10.3.2	连接新的系统调用	123
9.4.12	t-shell 的运行效果	113	10.3.3	重新编译 Linux 0.01 的内核	123
	本章小结	113		本章小结	124
第 10 章	Linux 0.01 系统调用的实现	114			
10.1	Linux 系统调用概述	114			

Linux 和 GNU 简介

知识点

- 自由的天地：GNU 和 Linux
- 在乐趣中获得成功
- GNU 的开发工具

GNU 的主要精神是软件源代码应该自由流通。只有开放了软件的源代码，才能让软件开发者自由地与其他使用者和开发人员进行充分交流，共同创造出优秀的软件。

Linux 得以快速发展很大程度上可以归功于 GNU 的主要精神。希望读者在学习 Linux 内核的同时，领会到 GNU 精神的魅力所在。

Linux 所有的源代码都可以被使用者轻易获得，并进行任意的研究和修改。自由的 Linux 为计算机爱好者提供了学习操作系统设计的最好教材。本书将深入地分析 Linux 0.01 的源代码，同时借助 GNU 提供的开发工具指导读者进行操作系统设计实验，使读者在提高 Linux 下编程水平的同时，能够快速掌握操作系统设计的基本原理和技术。

自由软件提供了大量的开发工具，如 GCC、make、nasm、ld 等，使用 GNU 丰富的开发工具，学习和编写操作系统是非常方便的。

1.1 自由的天地：GNU 和 Linux

GNU 是由自由软件基金会(Free Software Foundation, FSF)的董事长 Richard M. Stallman(RMS)于 1984 年发起的，至今已经有 20 多年的历史了。

GNU 是 GNU's Not UNIX 的缩写。GNU 的创始人 Stallman 认为 UNIX 虽然不是最好的操作系统，但是至少不会太差，而他自信有能力把 UNIX 不足的地方加以改进，使它成为一个优秀的操作系统。Stallman 把这个更优秀的操作系统命名为 GNU，开发 GNU 的目的就是为了让所有计算机用户都可以自由地获得这个系统及其源代码，并且可以相互自由拷贝。因此，在使用 GNU 软件时可以理直气壮地说自己使用的是“正版软件”。

在 GNU Manifesto (GNU 宣言) 中对 GNU 的精神进行了阐述: 软件的源代码应该自由流通, 软件开发者应该做的不是把源代码据为己有, 赚取发行可执行文件的金钱, 而是应该赚取整合与服务费用。因为源代码自由流通的软件才能让软件的质量提高, 让软件开发人员可以自由地与他人交换心得, 不受知识产权的约束。

为了保证 GNU Manifesto 精神的实施, GNU 制定的 GPL (The GNU General Public License, GNU 通用公共准许证) 先依照著作权法获得 GNU 软件的版权, 再通过 GPL 释放此权力给所有使用者, 只要用户遵守 GPL, 不把源代码以及自己对源代码所作的修改据为己有, 就拥有使用 GPL 软件的权力。

1.1.1 热爱和享受自由

自由软件发展至今已经成为计算机界一个重要的组成部分。通过不懈的发展, “自由软件”越来越显示出它所具有的强大力量, 使其成为计算机技术发展的重要推动力。

使用自由软件可以获得前所未有的自由, 包括: 使用自由 (可以不受任何限制来使用软件)、研究自由 (可以研究软件运作方式, 并使其适合个人需要)、散布自由 (可以自由地复制此软件并散布给他人)、改良自由 (可以自行改良软件并散布改良后的版本以使全体用户受益)。

作为自由软件代表之一的 Linux, 已经在世界无数的服务器上运行, 为公司、个人提供强大的计算服务。Linux 现在几乎可以胜任从桌面计算到企业服务的各个领域, 成为强大、自由、免费的操作系统。

自由软件为什么会有如此大的魅力呢?

(1) 自由软件的适应范围要比商业软件广得多。自由软件和传统商业软件之间最显著的差异在于: 自由软件鼓励拷贝, 允许研究、改良。只要有人需要某种功能, 就有人把这种功能加入到程序里。自由软件也往往具有很强的可移植性, 可以非常容易地在不同的软硬件平台上使用。而商业软件在功能性和可移植性上都无法和自由软件相比, 缺乏源代码也直接导致了用户对商业软件难以进行维护和改进。

(2) 自由软件的全部或部分能够被随意地利用、改进、再次发行。自由软件的使用者可以修改发现的 bug 或者加入自己需要的功能, 或者移植给其他系统。自由软件通常在世界范围内总是由顶级高手进行维护。在软件的使用过程中, 如果报告一个 bug, 很快就有人解决这个问题, 提出一个新的修改。通过互联网的交流和沟通, 对一些他人发现的 bug, 感兴趣的程序员也可以通过源代码的分析来解决问题。借助于互联网的交流, 联合全世界无数对自由软件热爱的开发人员, 自由软件的升级和更新通常比商业软件快得多。正因为如此, 自由软件相比商业软件具有更少的严重 bug 和缺陷。

对于自由软件来说, 使用者能够自己改进程序后再次发行, 这样软件的功能就会越来越强大, 衍生出来的软件会使更多的人受益。比如, 对于 Linux 操作系统, 中国的程序员完全可以在其内核中加入中文处理能力, 将 Linux 改造成为“纯粹”的中文自由操作系统。商业软件没有这个优点, 商业软件的升级版本可能需要付费, 或者根本就不会有使用者所需要的那种升级版本诞生。

自由软件具有强大的生命力, 这是由它的开放性决定的。一个自由软件, 哪怕只剩一个人喜欢, 他都可以自己来维护这个程序的生存, 适应自己的需要, 说不定以后还会有更多的人对这个程序产

生兴趣。

(3) 自由软件有良好的社会作用。它的一切工作原理都是公开的，这体现了尊重科学、信息公开、共同进步的良好风尚。它能够被随意地拷贝给需要它的人们使用，一个理解自由软件思想的人会更加关爱社会、乐于助人，对于改善整个社会风气都有很大的好处。

1.1.2 神奇的 Linux

1991 年 8 月，一位来自芬兰赫尔辛基大学的年轻人 Linus Benedict Torvalds（李纳斯·托沃兹）对外发布了一套全新的操作系统：Linux。

Linus 是一名大学生，为了实习使用了著名计算机科学家 Andrew S. Tanenbaum 教授开发的 Minix（一套功能简单、易学易懂的 UNIX 操作系统，可以在 8086 上运行，后来也支持 80386，在一些 PC 机平台上非常流行），Linus 购买了一台 386 微机，但是他发现 Minix 的功能还很不完善，于是他决心自己写一个保护模式下的操作系统，这就是 Linux 的原型。

最开始的 Linux 是用汇编语言编写的，主要工作是用来处理 80386 保护模式，按照 Linus 本人的描述，刚开始的时候是这样的：

“最开始的确是一次痛苦的航行，但是我终于可以拥有自己的一些设备驱动程序了，并且排错也变得更容易了，我开始使用 C 语言来开发程序，这大大加快了开发速度，我开始担心我发的誓言：‘作一个比 Minix 更好的 Minix’，我梦想有一天我能在 Linux 下重新编译 GCC……”

“我花了两个月来进行基本的设置工作，直到我拥有了一个磁盘驱动程序（有很多错误，但碰巧能在我的机器上工作）和一个小小的文件系统，这就是我的第 0.01 版（大约是 1991 年 8 月下旬的事情），它并不完善，连软盘驱动器的驱动程序都没有，什么事情也做不了，但是我已经被它吸引住了，除非我能放弃使用 Minix，不然我不会停止改进它。”

上述文字引自《乐者为王》（英文名为 *Just For Fun*）——自由软件 Linux 之父李纳斯·托沃兹自述，中国青年出版社 2001 年 7 月出版。

1991 年 10 月 5 日，Linus 发布了 Linux 的第一个“正式”版本：0.02 版，这时的 Linux 可以运行 bash（GNU 的一个 UNIX shell 程序）、GCC（GNU 的 C 编译器），它几乎还是什么事情都做不了，主要的注意力被集中在系统核心的开发工作上，没有人去注意用户支持、文档工作、版本发布等其他东西。

最开始的 Linux 版本被放置到一个 FTP 服务器上供大家自由下载，FTP 服务器的管理员认为这是 Linus 的 Minix，因而就建了一个 Linux 目录来存放这些文件，于是 Linux 这个名字就传开了，如今已经成了这个伟大的操作系统的约定名称。

Linus 在 USENET 讨论区 comp.os.minix 首先发布了下面这条消息：

“大家可曾渴望 Minix-1.1 会有这样美好的一天：每个人可以自己编写驱动程序，但是可能我们没有发现这样一个美妙的计划——可以自己修改操作系统以适应自己的需要？你是否对所有东西都在 Minix 上运行这一点感到沮丧？你是否没有找到一个业余时间可以干的好题目？下面这篇文章也许正是你所需要的。”

“如同我在一个月以前所提到的那样，我正在开发一个类似于 Minix 的基于 80386 的操作系统，它现在已经可以工作了（当然得看你怎么想），现在我将公布它的源代码，它是第 0.02 版本，但是

可以运行 bash、GCC、gnu-make、gnu-sed、compress 等。”

该文引自 Linus 在 USENET 讨论区 comp.os.minix 中发表 Linux 核心 0.02 时的说明。

在这以后，这个“娃娃”操作系统就以两个星期出一次修正版本的速度迅速成长，在版本 0.03 之后 Linus 将版本号迅速提高到 0.10，这时候更多的人开始在这个系统上工作。在几次修正之后 Linus 将版本号提高到 0.95，这表明他希望这个系统迅速成为一个“正式”的操作系统，这时候是 1992 年，但是直到一年半之后，Linux 的系统核心版本仍然是 0.99，已经非常接近 1.0 了。



提示

版本号的命名规范各有不同，GNU 风格的版本号管理策略是，项目初版本时，版本号可以为 0.1.0，也可以为 1.0.0；当项目在原有的基础上增加了部分功能时，主版本号不变，次版本号加 1，修正版本号复位为 0，可以被忽略，如上面的 0.02；当项目在进行了局部修改或 bug 修正时，主版本号和次版本号都不变，修正版本号加 1，如 0.1.1；当项目在进行了重大修改或局部修正累积较多，而导致项目整体发生全局变化时，主版本号加 1，如 1.0。

Linux 终于在 1994 年 3 月 14 日发布了它的第一个正式版本 1.0 版。目前 Linux 已经是一个功能强大的操作系统了，在它上面可以运行无数的自由软件，借助 Linux 可以完成几乎所有其他商用操作系统才能完成的工作，这就是自由的 Linux。

如图 1-1 所示是 www.linux.org 网站（Linux 官方网站）上的 Linux 吉祥物——一只可爱的小企鹅（起因是因为 Linus 是芬兰人，因而挑选企鹅作为吉祥物）。



图 1-1 Linux 的吉祥物

Linux 操作系统是一个类 UNIX 操作系统，具有以下特色：

- 多用户和多任务支持。
- 具有方便的命令行、用户界面和优秀的用户图形界面。
- 设备独立性，内核具有高度适应能力。
- 丰富的网络功能。
- 可靠的系统安全。
- 良好的可移植性。
- 丰富的应用软件。
- 良好的开放性，可以免费获得源代码。

正因为 Linux 具有如此多的优点，所以越来越多的计算机爱好者投入到学习 Linux 的行列中来。对于软件工程师而言，Linux 作为一个功能强大、源码开放的操作系统，是学习操作系统设计的最好教材。

目前，Linux 内核的最新版本是 2.6.30。2.6.30 的内核是一个完善的操作系统内核，具有丰富

的功能。但是 2.6.30 内核的源代码数量也是惊人的，有数百万行之巨。



注意

Linux 核心源代码可以在以下站点下载：

<http://www.kernel.org/pub/linux/>

<http://www.de.kernel.org/pub/linux/>

如表 1-1 所示是不同版本的 Linux 的核心文件数量和源代码数量，如图 1-2 和图 1-3 所示是不同版本的 Linux 核心源代码比较图。

表 1-1 不同版本的 Linux 的核心文件数量和源代码行数

Linux 内核版本号	文件数量 (个)	源代码行数 (行)
0.01	76	8413
0.12	99	15486
0.97	187	38928
1.00	487	165165
2.0.1	1643	686201
2.4.22	10302	4686534

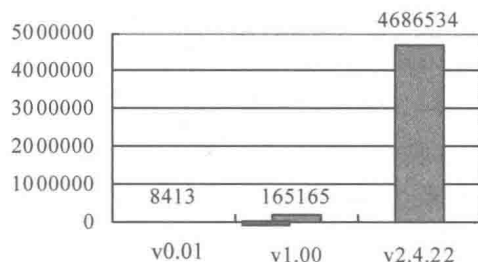
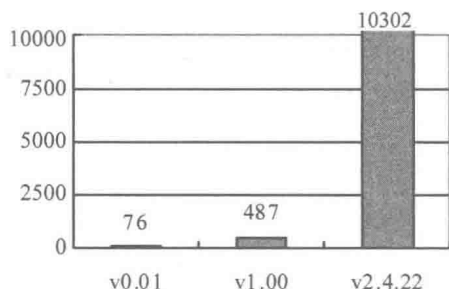


图 1-2 不同版本核心源代码的文件数量

图 1-3 不同版本核心源代码的源代码行数

从表 1-1 和图 1-2、图 1-3 中可以看出，Linux 核心源代码的文件数量和代码的行数随着版本的增长而飞速增加。对于 Linux 0.01 而言，文件数量是 76 个，所有源代码只有 8413 行。而 Linux 2.4.22 核心有文件 1 万多个，源代码行数有 468 万行之多。

对于有接近 500 万行源文件的 Linux 2.4.22 核心而言，要学习它是非常困难的。因为在这个核心中功能太多了，使学习者难以把握操作系统最主要的精髓。

而 Linux 0.01 虽然是 Linux 的第一个发行版本，但是却基本具备了操作系统中最重要的组成部分。同时，Linux 0.01 只有 1 万行左右的代码，对于初学者而言，学习起来相对简单多了。因此，本书以 Linux 0.01 核心为背景来介绍操作系统的设计。

1.2 在乐趣中获得成功：一起来玩操作系统

Linus 编写的 Linux 是最著名的自由软件之一，如今已经发展成为能够与 UNIX 和微软的

Windows 抗衡的自由软件操作系统。Linux 成为了自由软件的明星，现在不仅仅是计算机爱好者在研究和使用的 Linux，全世界几乎所有著名的软件公司（包括微软）都在研究 Linux。Linux 正在成为全世界计算机界的明星，而 Linus 也成为这个自由软件时代计算机爱好者的新的偶像。

如果说是商业利益的驱动成就了比尔·盖茨，那么 Linus 是因为追求电脑“乐趣”而无意中获得了成功。1997 年李纳斯·托沃兹从芬兰移居到硅谷，出任 Transmeta 公司首席软件科学家。李纳斯在《乐者为王》一书中，自述了他的成功之路。

1.2.1 十一岁开始编程

Linus 的外公是赫尔辛基大学的一位统计学教授。1981 年 Linus 11 岁时，外公抱回来一台新的 Commodore VIC-20 计算机。这种计算机不具备开发商业程序的条件，它能做的唯一事情就是用 BASIC 语言编程。Linus 的计算机生涯就是从这台计算机开始的。

通过使用 Commodore VIC-20 计算机，Linus 把键盘玩得很溜，然后 Linus 开始阅读电脑操作手册，并尝试将里面的示范程序（Example Program）输入进去。手册里有一些简单游戏的示范程序，如屏幕上会出现一个人横穿走过的图像，还可以稍作修改，以显示出各种不同的背景颜色，这些程序都大大地激发了 Linus 的兴趣。

然后 Linus 开始自己写程序。Linus 编写的第一个程序与其他人编的没什么不同，屏幕上显示出一行又一行的“HELLO”，直到厌烦而中止它。

这是 Linus 的第一步，也是许多人的最后一步。

但是，兴趣让 Linus 迈出了第二步和后续的无数多步。

Linus 的妹妹萨拉让 Linus 对这个程序作了修改，从而产生了这个程序的第二个版本，屏幕上显示的不再是“HELLO”的字样，而是无休无止的“SARA IS THE BEST”（“萨拉是最棒的”）。

在兴趣的驱动下，Linus 没有停下来，而是每写完一个程序就再去编写下一个程序。Linus 用零花钱购买电脑杂志，从中找到新的程序设计中的兴趣。就这样，一点一点地，Linus 逐渐走进了计算机的世界。

3 年过去了，当其他孩子在外面踢足球的时候，Linus 却觉得电脑更加有意思，机器本身就是一个自由的世界。计算机并没有想象中的那么复杂，Linus 在还是孩子的时候，就经常打开计算机的盖子自己动手修理。或许，计算机对于孩子而言，是一种很快乐的玩具。

1.2.2 一种操作系统的诞生

在兴趣的驱使下，Linus 上大学后第一年并没有选择相应的专业，计算机成了他的主修课。那时在整个赫尔辛基大学，连 Linus 在内，希望主修计算机的瑞典学生只有两个。

每个人都会有一本改变其一生的书籍，把 Linus 推向生命高峰的书是 Andrew S. Tanenbaum 写的《操作系统：设计和实现》。在这本书中，阿姆斯特丹的大学教授 Tanenbaum 讨论了 Minix 操作系统，那是他为 UNIX 撰写的教学辅助软件。读完此书，了解到 UNIX 背后的理念以及那个强大、简洁、漂亮的操作系统所能做到的事情后，Linus 便决定弄一台机器来“玩玩”UNIX。

赫尔辛基大学第一次拥有 UNIX 是在 1990 年秋季开学的时候。UNIX 的独到之处在于“优美”。可能很难理解使用优美来形容一个计算机软件，但是当理解了 UNIX 之后，就会发现 UNIX 的思想

和理念是简洁和优美的，是以后无数操作系统设计的典范。

1991 年，Linus 在赫尔辛基理工大学参加一个学术报告，演讲者理查德·斯多曼是个自由软件的鼓吹者。这是 Linus 第一次见到的典型“黑客”形象：留着长发，蓄着长胡子。这年，Linus 用圣诞节和生日得到的钱买了一台价值不菲的“组装”计算机：4MB 内存、33MHz 80386 CPU、14 英寸显示器。

Linus 购买的计算机本来运行的是 DOS 操作系统，但是 Linus 却不准备使用 DOS。因为，DOS 是一个太简单的操作系统，缺乏现代操作系统的基本要素。例如 DOS 不支持多任务和多用户。

安装 Minix 并不简单，Linus 用 16 张软盘把 Minix 装入计算机，并且使用了将近一个月时间，通过把需要的软件下载、安装、调试，“玩”了很久，才使这个系统完全变成了 Linus 需要的系统。

撰写 Minix 的 Andrew S. Tanenbaum 希望 Minix 是一个教学工具，因此 Minix 在许多设计方面并不是非常有利于真正的使用。在 Linus 购买这台计算机之前，Linus 就一直在网上跟踪 Minix 的消息，所以从一开始 Linus 就使用它的升级版。但 Minix 有一些性能令 Linus 很不满意，其中最大的失望是终端仿真（Terminal Emulation）。仿真很重要，因为 Linus 只能依赖这个程序才能让 Linus 家里的计算机登录到大学的计算机中。每当 Linus 拨电话接通大学的计算机，使用强大的 UNIX 工作或仅仅是上网时，都得使用终端仿真程序。

于是 Linus 决定自己开始做一个项目，制作自己的终端仿真程序。Linus 不想在 Minix 下做这个项目，而是想在硬件水平上完成它。为了不基于 Minix 来开发这个终端仿真程序，Linus 不得不从 BIOS 开始，BIOS 是计算机启动的早期 ROM 编码，它可以读软盘和硬盘，所以这次 Linus 是在软盘上操作。

80386 在刚启动的时候，是工作在“常规模式”下的，为了充分利用 80386 的 32 位能力，Linus 只得让 CPU 进入“保护模式”。

为了完成这个“项目”，Linus 设计了两条独立的线程：一条线程从调制解调器读出数据，然后在显示器上显示；另一条线程从键盘上读出数据，然后写入调制解调器。这样就会在两条线程上运行两条管道。Linus 最早的试验程序是使用一个线程将字母 A 写到显示器上，另一个线程写字母 B。Linus 把这个编入程序，让其在几秒钟之内出现若干次。在定时器的帮助下，使这个程序这样不停地运转：显示器上先出现一连串的字母 A，然后突然之间，转变成一连串的字母 B，这表明 Linus 所做的任务转换是可行的。最终 Linus 能改变由一连串 A 和一连串 B 组成的两个线程，从而使数据一个读自调制解调器，再写入显示器；另一个读自键盘，再写入调制解调器。这使 Linus 终于有了自己的终端仿真程序。

每当 Linus 想读新闻时，Linus 就运行自己的程序，把自己的软盘插进去，重新启动机器，这样就能从大学的计算机里读到所需要的新闻。倘若想要改进这个终端仿真程序，就得启动 Minix，使用 Minix 来对终端仿真程序进行修改。

Linux 操作系统就是这样诞生的。

现在，Linux 已经变成了具有强大吸引力的操作平台，全球大约已有数以千万计的计算机正在运行 Linux 系统。

1.3 GNU 的开发工具

GNU 拥有丰富的开发工具，例如：

- Vim：功能强大的编辑环境。
- GCC：性能优越，是通用的 C/C++、FORTRAN 的编译器。
- Kdevelop：KDE 集成开发工具。
- Cygwin：Windows 下的 GNU 开发环境。

同时，GNU 还拥有强大的编译自动化工具 `make`、汇编程序 `nasm`、连接器 `ld` 等。使用 GNU 丰富的开发工具，完全可以完成开发一个操作系统的所有工作。

本章小结

GNU 是一个伟大的事业。因为 GNU 的出现，才有机会能够深入地阅读和体会世界上最优秀软件的核心和精髓。

Linux 作为 GNU 中最优秀的软件之一，它的诞生是兴趣驱动的。兴趣可以创造一切。本书的目的也是让读者在乐趣中使用自由的软件，学习 Linux 编程的思想和技术。

自由软件提供了大量的开发工具，使用这些免费的开发工具可以方便地进行操作系统的开发。