

# Linux

内核版本为0.95版  
代码量不超过2万行

对Linux内核源代码进行 完全剖析 逐行注释

# Linux 权威 内核注释

读者能够深刻理解操作系统的基本功能和实现方式

孙更新 宾 晟 孙海伦 编著

中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

**Linux**



# Linux 内核注释

孙更新 宾 晟 孙海伦 编著

**中国铁道出版社**  
CHINA RAILWAY PUBLISHING HOUSE

## 内 容 简 介

本书通过对完整的 0.95 版本 Linux 内核源代码进行完全剖析和逐行注释,使得读者能够在较短时间内深刻理解 Linux 操作系统的基本功能和实现方式,为深入研究 Linux 操作系统打下坚实基础。作为 Linux 早期内核版本的 0.95 版本,其内核是目前所使用的较新内核的精简版,代码量比较小,却包括了几乎所有的基本功能原理的内容,通过阅读源代码能够使读者快速理解内核工作的基本原理。

本书共分 10 章,全面系统地介绍了 Linux 内核发展历史和版本变化,详细说明了 Linux 运行的硬件基础和 Linux 内核体系结构,完全剖析和注释了 Linux 内存管理、中断处理与系统调用、进程控制、块设备驱动程序、字符设备驱动程序、文件系统、引导启动程序、初始化程序,最后总结性地介绍了继续研究 Linux 系统的方法和着手点。

本书适合 Linux 爱好者学习内核工作原理使用,也可作为高校计算机专业学生学习操作系统课程的辅助和实验教材,还可供一般技术人员作为开发嵌入式系统的参考书使用。

### 图书在版编目(CIP)数据

Linux 内核注释/孙更新,宾晟,孙海伦编著.--

北京:中国铁道出版社,2011.12

ISBN 978-7-113-13019-0

I. ①L… II. ①孙…②宾…③孙… III. ①

Linux 操作系统 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2011)第 095565 号

书 名: Linux 内核注释

作 者: 孙更新 宾 晟 孙海伦 编著

---

责任编辑: 苏 茜

读者热线电话: 010-63560056

编辑助理: 贾淑媛

封面设计: 张 丽

责任印制: 李 佳

---

出版发行: 中国铁道出版社(北京市西城区右安门西街 8 号 邮政编码: 100054)

印 刷: 北京新魏印刷厂

版 次: 2011 年 12 月第 1 版

2011 年 12 月第 1 次印刷

开 本: 787mm × 1092mm 1/16

印张: 28.5 字数: 675 千

书 号: ISBN 978-7-113-13019-0

定 价: 69.00 元

---

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社发行部联系调换。

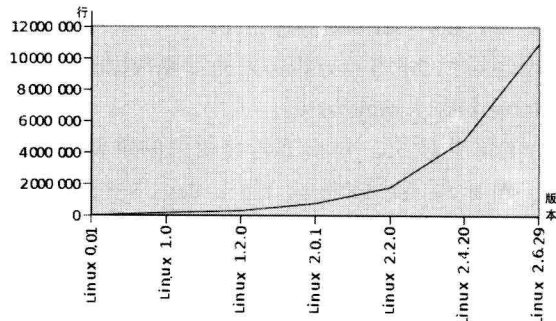
# 前言

研究 Linux 内核可以彻底理解 Linux 的内部构造及其基本工作原理。本书通过对完整的 Linux 内核源代码进行完全剖析和逐行注释，使得读者能够深刻理解操作系统的基本功能和实现方式，最终达到对 Linux 操作系统基本工作原理理解的目的。

本书是一本关于 Linux 操作系统内核基本原理的入门级读物，主要面向具有一定编程技能和熟悉 Linux 系统，同时又想要了解 Linux 操作系统内核工作原理和实际代码实现的读者。这部分读者人数众多，但满足其需求的相关书籍却很少，本书就是这部分读者的最佳选择之一。

## 内核版本选择

目前，Linux 内核的源代码量非常大，有数百万行之多，因此无法对这些版本进行完全地注释和说明，仅能对其有选择性地说明，这将会给读者一种盲人摸象的感觉，而无法给予读者对实际 Linux 内核清晰完整的理解。早期内核源代码则是目前所使用的较新内核的精简版，代码量比较小，并且包括了几乎所有的基本功能原理的内容。选择 Linux 早期内核版本进行剖析，可以避免实际内核功能实现的复杂性，又能透彻地说明问题，使读者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。下图是对一些主要 Linux 内核版本源代码量的统计。



通过大量的调查和对内核版本的详细比对，本书选择了与目前 Linux 内核基本功能较为相近，同时代码量又比较小的 0.95 版内核（代码量不超过 2 万行）作为分析的对象，因此本书完全可以对 Linux 内核的基本功能解释和注释清楚。

## 读者应该具备的基础知识

阅读本书的读者应该具有以下一些基础知识：

- 掌握 C 语言的相关知识。
- 掌握有关 80×86 处理器结构和编程的知识。
- 掌握有关 80×86 硬件体系结构和接口编程的知识。
- 对 Linux 比较熟悉，具备使用 Linux 操作系统的初级技能。如果读者还具有操作系统设计相关经验或其他计算机科学的概念则更佳。

## 面向读者群体

本书适合 Linux 爱好者学习内核工作原理使用，也可作为高校计算机专业学生学习操作系统课程的辅助和实验教材，还可供一般技术人员作为开发嵌入式系统的参考书使用。

## 本书的组织形式

本书各章之间组织有序，内容循序渐进又相对独立。对于不了解 Linux 内核的读者，可以由浅入深逐步理解内核的工作原理和实现方式。对内核基本情况已经有所了解的读者，完全可以按照自己的实际需要直接阅读相关章节。

第 1 章对 Linux 操作系统原理进行了简单介绍，主要包括 Linux 操作系统的内核构成及操作系统的特征，Linux 操作系统的诞生、成长和发展，然后介绍了阅读和研究 Linux 内核的重要意义等内容。

第 2 章讨论了引导与初始化程序，分析并注释了程序 `bootsect.s`、`setup.s`、`head.s`、`main.c`。

第 3 章分析并注释了 Linux 内核的所有头文件。

第 4 章讨论了中断处理和系统调用的总体功能，介绍了与其相关的重要数据结构，分析并注释了头文件 `unistd.h` 和 `sys.h` 以及程序 `traps.c`、`sys.c`、`system_call.s`、`asm.s`。

第 5 章讨论了进程控制，介绍了与其相关的重要数据结构，分析并注释了程序 `fork.c`、`sched.c`、`signal.c`。

第 6 章讨论了字符设备驱动程序，介绍了与其相关的重要数据结构，分析并注释了程序 `console.c`、`keyboard.s`、`pty.c`、`rs_io.s`、`serial.c`、`tty_io.c`。

第 7 章讨论了块设备驱动程序，介绍了与其相关的重要数据结构，分析并注释了头文件 `blk.h` 以及程序 `floppy.c`、`hd.c`、`ll_rw_blk.c`、`ramdisk.c`。

第 8 章讨论了 Linux 的文件系统，介绍了与其相关的重要数据结构，分析并注释了程序 `buffer.c`、`inode.c`、`namei.c`、`super.c`、`file_table.c`、`block_dev.c`、`pipe.c`、`char_dev.c`、`read_write.c`、`open.c`、`stat.c`、`fcntl.c`、`ioctl.c`、`select.c`、Minix 的 `bitmap.c`、Minix 的 `inode.c`、Minix 的 `truncate.c`、Minix 的 `namei.c`、Minix 的 `file_dev.c`、Minix 的 `minix_op.c`。

第 9 章讨论了 Linux 的内存管理，分析并注释了头文件 `mm.h` 和 `memory.h` 以及程序 `memory.c`、`swap.c`。

第 10 章介绍了 Linux 内核的建造工具与编程环境，重点分析并注释了生成内核磁盘映像文件的工具程序 `buidl.c` 和如何在 Linux 下进行开发。

## 创作群体

本书由孙更新、宾晟、孙海伦编写，下面人员对本书的编写提出过宝贵意见并参与了部分编写工作，他们是张杰、徐作英、邵长恒、宋小宁、吴洪飞、刘树一、刘丽、孔波、颜浩、贺萍、于瑞菊、王亚平、任晓雪、李琼、高宝玉、孙军、徐凯。

由于时间仓促，加之水平有限，书中的缺点和不足之处在所难免，敬请读者批评指正。如果读者在阅读本书的时候出现任何疑问，还可以发送电子邮件及时与我们联系，我们会尽快给予答复。我们的邮箱是 [suqian@tqbooks.net](mailto:suqian@tqbooks.net)。

# 目 录

<b>第 1 章 Linux 操作系统原理</b> .....	<b>1</b>
1.1 Linux 操作系统简介 .....	1
1.1.1 Linux 内核的构成 .....	4
1.1.2 Linux 操作系统的特征 .....	5
1.2 Linux 内核概述 .....	6
1.2.1 Linux 内核的版本 .....	6
1.2.2 研究 Linux 内核的意义 .....	9
1.3 Linux 运行的硬件基础 .....	10
1.3.1 i386 寄存器和系统指令 .....	10
1.3.2 分段机制 .....	15
1.3.3 分页机制 .....	23
1.4 Linux 内核体系结构 .....	26
1.4.1 Linux 内核体系结构及模式 .....	27
1.4.2 Linux 内核源代码目录结构 .....	28
1.4.3 Linux 内核的编译 .....	32
<b>第 2 章 引导启动与初始化程序</b> .....	<b>36</b>
2.1 引导启动程序 (boot) .....	36
2.1.1 总体功能 .....	36
2.1.2 bootsect.s 文件 .....	37
2.1.3 setup.s 文件 .....	42
2.1.4 head.s 文件 .....	47
2.2 初始化程序 (init) .....	51
2.2.1 概述 .....	51
2.2.2 main.c 文件 .....	51
<b>第 3 章 头文件</b> .....	<b>57</b>
3.1 include/目录下的文件 .....	57
3.2 a.out.h 文件 .....	57
3.3 const.h 文件 .....	61
3.4 ctype.h 文件 .....	62
3.5 errno.h 文件 .....	63
3.6 fcntl.h 文件 .....	64
3.7 signal.h 文件 .....	65
3.8 stdarg.h 文件 .....	67
3.9 stddef.h 文件 .....	68



3.10	string.h 文件 .....	68
3.11	termios.h 文件 .....	77
3.12	time.h 文件 .....	82
3.13	unistd.h 文件 .....	83
3.14	utime.h 文件 .....	89
3.15	include/asm 目录下的文件 .....	90
3.16	io.h 文件 .....	90
3.17	memory.h 文件 .....	91
3.18	segment.h 文件 .....	91
3.19	system.h 文件 .....	93
3.20	include/linux/目录下的文件 .....	95
3.21	config.h 文件 .....	95
3.22	fdreg.h 头文件 .....	96
3.22.1	功能描述 .....	96
3.22.2	代码注释 .....	97
3.23	fs.h 文件 .....	99
3.24	hdreg.h 文件 .....	103
3.25	head.h 文件 .....	105
3.26	kernel.h 文件 .....	105
3.26.1	功能描述 .....	105
3.26.2	代码注释 .....	105
3.27	mm.h 文件 .....	106
3.28	sched.h 文件 .....	107
3.29	sys.h 文件 .....	112
3.30	tty.h 文件 .....	114
3.31	math_emu.h 文件 .....	116
3.31.1	功能描述 .....	116
3.31.2	代码注释 .....	116
3.32	minix_fs.h 文件 .....	120
3.33	timer.h 文件 .....	121
3.34	include/sys/目录下的文件 .....	122
3.35	param.h 文件 .....	122
3.36	ptrace.h 文件 .....	122
3.37	resource.h 文件 .....	124
3.38	stat.h 文件 .....	125
3.39	time.h 文件 .....	126
3.40	times.h 文件 .....	127
3.41	types.h 文件 .....	128
3.42	utsname.h 文件 .....	129
3.43	wait.h 文件 .....	129

<b>第 4 章 中断处理与系统调用</b> .....	<b>130</b>
4.1 总体功能.....	130
4.1.1 中断处理.....	130
4.1.2 系统调用.....	131
4.2 重要数据结构.....	132
4.3 sys.h 文件.....	132
4.4 unistd.h 文件.....	134
4.5 IDT 初始化——_set_gate.....	141
4.6 traps.c 文件.....	141
4.7 sys.c 文件.....	146
4.8 system_call.s 文件.....	156
4.9 asm.s 文件.....	162
4.10 ptrace.c 文件.....	165
<b>第 5 章 进程控制</b> .....	<b>172</b>
5.1 总体概述.....	172
5.2 重要数据结构.....	174
5.2.1 task_struct.....	174
5.2.2 tss_struct.....	175
5.2.3 i387_struct.....	176
5.2.4 task_union, init_task, current.....	176
5.2.5 sigaction.....	176
5.2.6 timer_list.....	176
5.3 fork.c 文件.....	177
5.3.1 功能描述.....	177
5.3.2 代码注释.....	177
5.4 sched.c 文件.....	180
5.4.1 功能描述.....	181
5.4.2 代码注释.....	181
5.5 signal.c 文件.....	190
5.5.1 功能描述.....	190
5.5.2 代码注释.....	190
5.6 exit.c 文件.....	195
5.6.1 功能描述.....	195
5.6.2 代码注释.....	195
5.7 vsprintf.c 文件.....	203
5.8 printk.c 文件.....	208
5.9 panic.c 文件.....	209





<b>第 6 章 字符设备驱动程序</b> .....	<b>210</b>
6.1 总体功能 .....	210
6.1.1 Linux 支持的终端设备类型 .....	210
6.1.2 Linux 字符设备驱动程序 .....	212
6.1.3 字符设备驱动程序功能 .....	212
6.2 重要数据结构 .....	213
6.3 console.c 文件 .....	214
6.4 keyboard.s 文件 .....	238
6.5 pty.c 文件 .....	252
6.6 rs_io.s 文件 .....	253
6.7 serial.c 文件 .....	256
6.8 tty_io.c 文件 .....	258
6.8.1 功能描述 .....	258
6.8.2 代码注释 .....	259
6.9 tty_ioctl.c 文件 .....	268
<b>第 7 章 块设备驱动程序</b> .....	<b>276</b>
7.1 总体功能 .....	276
7.1.1 Linux 设备管理 .....	276
7.1.2 Linux 块设备驱动程序 .....	277
7.1.3 块设备驱动程序功能 .....	278
7.2 重要数据结构 .....	279
7.3 blk.h 文件 .....	280
7.4 floppy.c 文件 .....	283
7.5 hd.c 文件 .....	293
7.5.1 功能描述 .....	293
7.5.2 代码注释 .....	294
7.6 ll_rw_blk.c 文件 .....	303
7.7 ramdisk.c 文件 .....	308
<b>第 8 章 文件系统</b> .....	<b>311</b>
8.1 总体功能 .....	311
8.1.1 VFS 虚拟文件系统 .....	312
8.1.2 Minix 文件系统 .....	312
8.1.3 文件系统程序功能 .....	313
8.2 重要数据结构 .....	314
8.3 buffer.c 文件 .....	317
8.3.1 功能描述 .....	317
8.3.2 代码注释 .....	318
8.4 inode.c 文件 .....	326

8.4.1 功能描述.....	326
8.4.2 代码注释.....	327
8.5 namei.c 文件.....	332
8.5.1 功能描述.....	332
8.5.2 代码注释.....	333
8.6 super.c 文件.....	342
8.6.1 功能描述.....	342
8.6.2 代码注释.....	342
8.7 file_table.c 文件.....	349
8.8 block_dev.c 文件.....	349
8.8.1 功能描述.....	349
8.8.2 代码注释.....	349
8.9 pipe.c 文件.....	351
8.10 char_dev.c 文件.....	354
8.11 read_write.c 文件.....	358
8.12 open.c 文件.....	360
8.13 exec.c 文件.....	365
8.14 stat.c 文件.....	372
8.15 fcntl.c 文件.....	374
8.16 ioctl.c 文件.....	376
8.17 select.c 文件.....	377
8.18 Minix 的 bitmap.c 文件.....	383
8.19 Minix 的 inode.c 文件.....	387
8.20 Minix 的 truncate.c 文件.....	391
8.21 Minix 的 namei.c 文件.....	394
8.22 Minix 的 file_dev.c 文件.....	411
8.23 Minix 的 minix_op.c 文件.....	413
<b>第 9 章 内存管理.....</b>	<b>415</b>
9.1 总体功能描述.....	415
9.1.1 内存分页机制管理.....	415
9.1.2 Linux 中内存的使用.....	417
9.2 memory.c 文件.....	417
9.2.1 功能描述.....	417
9.2.2 代码分析.....	417
9.3 swap.c 文件.....	429
<b>第 10 章 Linux 内核编程环境.....</b>	<b>435</b>
10.1 建造工具介绍.....	435
10.1.1 总体功能.....	435



10.1.2	重要数据结构.....	435
10.1.3	build.c 文件.....	436
10.2	在 Redhat 9 下修改和编译 Linux 0.95 内核.....	439
10.2.1	源代码中注释语句的修改.....	439
10.2.2	源代码中 makefile 文件的修改.....	439
10.2.3	源代码中 align 值的修改.....	440
10.2.4	源代码中嵌入宏汇编程序的修改.....	440
10.2.5	源代码中汇编程序 C 变量的修改.....	440
10.2.6	在保护模式下调试显示函数.....	440
10.2.7	编译 Linux 0.95 内核.....	441
10.3	在 Bochs 上运行 Linux 0.95 系统.....	441
10.3.1	创建磁盘映像文件.....	441
10.3.2	访问磁盘映像文件.....	441
10.3.3	创建文件系统.....	442
10.3.4	在硬盘映像文件上建立根文件系统.....	442
10.3.5	使用硬盘映像上的根文件系统运行 Linux 0.95.....	443

# 第 1 章 Linux 操作系统原理

操作系统（Operating System, OS）是控制其他程序运行、管理系统资源并为用户提供操作界面的系统软件的集合。它是管理计算机硬件与软件资源的程序，同时也是计算机系统的内核与基石。操作系统理论在计算机科学中是历史悠久而又活跃的分支，而操作系统的设计与实现则是软件工业的基础与内核。操作系统本身负责诸如管理与配置内存、决定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务。

操作系统能够使计算机系统的所有资源最大限度地发挥作用，为用户提供方便、有效、友善的服务界面。操作系统是一个庞大的管理控制程序，大致包括 5 个方面的管理功能：进程与处理机管理、作业管理、存储管理、设备管理、文件管理。目前的操作系统种类繁多，不同类型计算机安装的操作系统有的简单有的复杂，从手机的嵌入式系统到超级电脑的大型操作系统，很难用单一标准统一分类。微型计算机上常见的操作系统包括 DOS、OS/2、UNIX、XENIX、Linux、Windows、NetWare 等。但所有的操作系统都具有并发性、共享性、虚拟性和不确定性四个基本特征。

## 1.1 Linux 操作系统简介

Linux 是当前的主流操作系统之一，因其开放源代码的特性而深受众多用户的青睐。

### 1. Linux 的诞生和发展

早期的操作系统是在 20 世纪 50 年代开发的，用来提供简单的开发体验，包括为 IBM 701 开发的 General Motors Operating System (GMOS) 以及为 IBM 709 开发的 FORTRAN Monitor System (FMS)。

在 20 世纪 60 年代，麻省理工大学 (Massachusetts Institute of Technology, MIT) 和一些公司为 GE-645 开发了一个名为 Multics (Multiplexed Information and Computing Service) 的实验性操作系统。这个操作系统的开发者之一 AT&T 后来退出了 Multics，并在 1970 年开发了自己的名为 Unics 的操作系统。与这个操作系统一同诞生的是 C 语言，然后他们使用 C 语言对操作系统进行了重写，使操作系统开发具有可移植性。

20 年后，为了教学的需要，荷兰计算机教授 Andrew S. Tanenbaum 创建了一个微内核版本的 UNIX，名为 MINIX (其含义为 minimal UNIX)，它可以在小型的个人计算机上运行。

Linux 操作系统于 1991 年诞生于一个芬兰大学生 Linus Torvalds 之手：1991 年 8 月，他在赫尔辛基大学学习操作系统课程时，由于不满足于使用教学用的操作系统 MINIX，于是着手开发一个简单的程序，并逐步开发了显示器、键盘和调制解调器的驱动程序，最后编写了磁盘驱动程序、文件系统等，一个操作系统的原型就这样形成了，后来它逐步发展为与 UNIX、Windows 并驾齐驱的实用操作系统。

Linux 操作系统是一种自由的 UNIX 类多用户、多任务操作系统，它可以运行在多种计算机平台上。自 1991 年 10 月 5 日，Linus 正式向外宣布 Linux 内核系统的诞生以来，它已经成为应用广泛、可靠性高，功能强大的计算机操作系统。借助于 GNU 计划、POSIX 标准、Internet，并经过世界各地计算机爱好者的共同努力，Linux 现已成为目前使用较多的一种 UNIX 类操作系统，并且使用人数还在迅猛增长。

Linux 操作系统内核的名字也是 Linux。Linux 操作系统是自由软件和开放源代码发展中最著



名的例子之一。严格来讲，Linux 这个词本身只表示 Linux 内核，但在实际上人们已经习惯了用 Linux 来形容整个基于 Linux 内核，并且使用 GNU 工程各种工具和数据库的操作系统（GNU/Linux）。基于这些组件的 Linux 软件被称为 Linux 发行版。一般来讲，一个 Linux 发行套件包含大量的软件，例如，软件开发工具、数据库、Web 服务器、X Window，桌面环境、办公套件等。Linux 内核最初是为 Intel 386 微处理器设计的。

现在 Linux 内核支持从个人电脑到大型主机甚至包括嵌入式系统在内的各种硬件设备。现在，Linux 已经成为了受到广泛关注和支持的一种操作系统，包括 IBM 和惠普在内的一些计算机业巨头也开始支持 Linux。很多人认为，和其他的商用 UNIX 操作系统以及微软的 Windows 系列操作系统相比，作为自由软件的 Linux 具有成本低、安全性高、更加可信赖的优势。

Linux 操作系统在短短的几年之内得到了非常迅猛的发展，这与 Linux 具有的良好特性是分不开的。Linux 包含了 UNIX 的全部功能和特性。简单地说，Linux 具有以下主要特性：

- **开放性：**指系统遵循世界标准规范，特别是遵循开放系统互连（OSI）国际标准。凡遵循国际标准所开发的硬件和软件，都能彼此兼容，可方便地实现互连。另外，源代码开放的 Linux 是免费的，使得获取 Linux 非常方便，而且使用 Linux 可节省费用。Linux 开放源代码，使用者能控制源代码，按照需要对部件混合搭配，建立自定义扩展。
- **多用户：**指系统资源可以被不同用户各自拥有使用，即每个用户对自己的资源（如文件、设备）有特定的权限，互不影响。
- **多任务：**多任务是现代计算机的最主要的特点之一，是指计算机同时执行多个程序，而且各个程序的运行互相独立。Linux 系统调度每一个进程平等地访问微处理器。
- **出色的速度性能：**Linux 可以连续运行数月、数年而无需重新启动，与 NT（经常死机）相比，这一点尤其突出。即使作为一种台式机操作系统，与许多用户非常熟悉的 UNIX 相比，它的性能也显得更为优秀。Linux 不大在意 CPU 的速度，它可以把处理器的性能发挥到极限（用户会发现，影响系统性能提高的限制因素主要是其总线和磁盘 I/O 的性能）。
- **良好的用户界面：**Linux 向用户提供了 3 种界面：用户命令界面、系统调用界面和图形用户界面。
- **提供了丰富的网络功能：**Linux 是在 Internet 基础上产生并发展起来的，因此，完善的内置网络是 Linux 的一大特点。Linux 在通信和网络功能方面优于其他操作系统。
- **可靠的系统安全：**Linux 采取了许多安全技术措施，包括对读、写进行权限控制、带保护的子系统、审计跟踪、核心授权等，这为网络多用户环境中的用户提供了必要的安全保障。
- **良好的可移植性：**可移植性是指将操作系统从一个平台转移到另一个平台时它仍然能按其自身方式运行的能力。Linux 是一种可移植的操作系统，能够在从微型计算机到大型计算机的任何环境和任何平台上运行。可移植性为运行 Linux 的不同计算机平台与其他任何机器进行准确而有效地通信提供了手段，不需要另外增加特殊和昂贵的通信接口。
- **具有标准兼容性：**Linux 是一个与 POSIX（Portable Operating System Interface）相兼容的操作系统，它所构成的子系统支持所有相关的 ANSI、ISO、IETF 和 W3C 业界标准。为了使 UNIX system V 和 BSD 上的程序能直接在 Linux 上运行，Linux 还增加了部分 system V 和 BSD 的系统接口，使 Linux 成为一个完善的 UNIX 程序开发系统。Linux 也符合 X/Open 标准，具有完全自由的 X-Window 实现。另外，Linux 在对工业标准的支持上做得非常好。由于各 Linux 发布厂商都能自由获取和接触 Linux 的源代码，各厂家发布的 Linux 仍然缺乏标准，不过这些差异非常小。它们的差异主要存在于所捆绑应用程序的版本、安装工具

的版本和各种系统文件所处的目录结构。

Linux 操作系统一般包含 4 个主要部分：内核、Shell、文件系统和应用程序，各部分层次结构如图 1-1 所示。内核、Shell 和文件系统形成了基本的操作系统结构。它们使得用户可以运行程序、管理文件并使用系统。

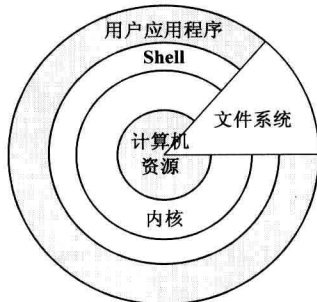


图 1-1 Linux 系统的组成

- **Linux 内核：**内核（Kernel）是系统的核心，实现操作系统的基本功能。  
在硬件方面：控制硬件设备，管理内存，提供硬件接口，处理基本 I/O。  
在软件方面：管理文件系统，为程序分配内存和 CPU 时间等。
- **Linux Shell：**Shell 是系统的用户界面，提供用户与内核进行交互操作的一种接口。Shell 是一个命令解释器，它解释由用户输入的命令并且把它们送到内核执行。Shell 编程语言具有普通编程语言的很多特点，用这种编程语言编写 Shell 程序与其他应用程序具有同样的效果。目前，常见的 Shell 有 Bourne Shell (sh)、Korn Shell (ksh)、C Shell (csh)、Bourne-again Shell (bash) 等。
- **Linux 应用程序：**标准的 Linux 系统都有一套称为应用程序的程序集，包括文本编辑器、编程语言、X-Window、办公套件、Internet 工具、数据库等。当然，还可以有用户自己编写的具有特定功能的应用程序。
- **Linux 文件系统：**文件系统是文件存放在磁盘等存储设备上的组织方法。通常是按照目录层次的方式进行组织。每个目录可以包括多个子目录以及文件，系统以“/”为根目录。系统中的所有数据都存储在文件系统中以使用户读取、查询和写入。Linux 能支持多种目前流行的文件系统，如 ext2、ext3、fat、vfat、ISO9660、nfs 等。

## 2. 常见的 Linux 操作系统

目前经常见到的 Linux 操作系统主要有 Red Hat、Ubuntu、SuSE、Red Flag、Turbo Linux、Fedora 等。

### (1) Red Hat

Red Hat Linux 最大的优点是它不仅已经是行业的标准，有大量为其开发的软件，而且它的产品稳定性也得到了广泛的认可。

1998 年 1 月，Red Hat 高级研发实验室成立，同年 Red Hat 5.0 获得了 InfoWorld 的操作系统奖项。4 月 Mozilla 代码发布，成为 Linux 图形界面上的王牌浏览器。同年 10 月，Intel 和 Netscape 宣布小额投资 Red Hat 软件，这被业界视作 Linux 获得商业认同的信号。12 月，IBM 发布了适用于 Linux 的文件系统 AFS 3.5 以及 Jikes Java 编辑器和 Secure Mailer 及 DB2 测试版。

1999 年 5 月，SGI 公司宣布向 Linux 移植其先进的 XFS 文件系统。7 月，IBM 启动对 Linux



的支持服务和发布了 Linux DB2，从此结束了 Linux 得不到支持服务的历史。

### (2) Ubuntu

Ubuntu 以快速上手以及长久的生存能力著称。快速上手，是指使用 Ubuntu 无需指导，花几分钟安装，立刻识别大部分硬件，隐藏 root（根）用户，避免有人弄乱它，接着预设配置，可使用户马上开始工作。Ubuntu 提供不错的开发者联合平台，OEM 顾客不断增长。

### (3) SuSE

欧洲市场占有率第一的发行版，其功能强大，界面华丽，安装 SuSE 可以选择显示英文或德文，它还有一套设定的程序叫 SaX，可以让使用者方便地设置，安装套件是 RPM 模式，所以要安装或删除程序都可以很方便。

### (4) Red Flag

这是由中科红旗软件技术公司推出的中文版本的 Linux，自从 2000 年 6 月成立以来该 Linux 在众多的中国 Linux 用户中占有一定的比例。Red Flag Linux 具有灵活的设计，将在嵌入式 Linux 领域为用户提供强大的技术支持服务。

### (5) Turbo Linux

它是一个安全、稳定、运行效率高、与硬件以及商业软件兼容性好的系统。该套件由 Pacific HiTech 公司开发，在日本占有一定的市场，从安装到使用都是日文的，在国内它与清华大学及研究机构合作研发了中文版本，掀起了一股 Linux 潮流。

### (6) Fedora

Fedora 是一个基于 Linux 的开放、创新、前瞻性的操作系统和平台，Fedora 项目是由 Red Hat 赞助，由开源社区与 Red Hat 工程师合作开发的项目统称。Fedora 的目标是推动自由和开源软件更快地进步。公开的论坛、开放的过程、快速的创新、精英和透明的管理，所有这些都为实现一个自由软件提供了最好的操作系统和平台。

## 1.1.1 Linux 内核的构成

Linux 内核由三个主要的子系统组成：进程调度(SCHED)、内存管理(MM)和文件系统(VFS)。

### 1. Linux 系统的进程调度

Linux 中的每个进程都分配有一个相对独立的虚拟地址空间。该虚拟空间分为两部分：用户空间包含了进程本身的代码和数据；内核空间包含了操作系统的代码和数据。

Linux 采用“有条件的可剥夺”调度方式。对于普通进程而言，当其时间片结束时，调度程序挑选出下一个处于 TASK\_RUNNING 状态的进程作为当前进程。对于实时进程，若其优先级足够高，则会从当前的运行进程中抢占 CPU 成为新的当前进程。发生强制调度时，若进程在用户空间中运行，就会直接被剥夺 CPU；若进程在内核空间中运行，即使迫切需要其放弃 CPU，也仍要等到从其系统空间返回前才被剥夺 CPU。

Linux 进程调度分为主动调度和被动调度两种方式：

主动调度随时都可以进行，Linux 内核可以通过 `schedule()` 启动一次调度，当然也可以将进程状态设置为 `TASK_INTERRUPTIBLE`，使得进程暂时放弃运行而进入睡眠状态；用户态中的程序可以通过 `pause()` 达到同样的目的；如果为这种暂时的睡眠放弃加上时间限制，内核态有 `schedule_timeout()`，用户态有 `nanosleep()` 用于实现此目的。这里需要注意的是，内核中这种主动放弃是不可见的，它隐藏在每一个可能受阻的系统调用中，例如 `open()`、`read()`、`select()` 等。

被动调度发生在系统调用返回的前夕、中断异常处理返回前、用户态处理软中断返回前。

自从 Linux 2.6 内核后, Linux 实现了抢占式内核, 即处于内核态的进程也可能被调度出去。例如, 一个进程正在内核态运行, 此时一个中断发生使另一个高权值进程就绪, 在中断处理程序结束之后, Linux 2.6 内核之前的版本会恢复原进程的运行, 直到该进程退出内核态才会引发调度程序; 而 Linux 2.6 抢占式内核, 在处理完中断后, 会立即引发调度, 切换到高权值进程。为支持内核代码可抢占, 在 Linux 2.6 版内核中通过采用禁止抢占的自旋锁来保护临界区。在释放自旋锁时, 同样会引发调度检查。而对那些长期持锁或禁止抢占的代码片段插入了抢占点, 此时检查调度需求, 以避免不合理的延迟发生。而在检查过程中, 调度进程很可能就会中止当前的进程来让另外一个进程运行, 只要新的进程不需要持有该锁。

## 2. Linux 系统的内存管理

Linux 内存管理采用分页机制来实现, 分页机制是应用程序通过软硬件协作来访问内存的一种方法, 其基本原理是将整个线性内存空间划分成若干页面, 在 80×86 体系结构中是以 4KB 为一页的内存页面。程序申请使用内存时, 就以一页为单位进行分配。

## 3. Linux 系统的文件系统

Linux 内核将文件系统的程序划分为两部分: 一部分实现了对虚拟文件系统的支持; 另一部分用于实现与 Minix 文件系统相关的操作, 该部分被单独放在一个名为 Minix 的文件夹内。

虚拟文件系统从功能上可以划分为 4 个部分。第一部分是高速缓冲区的管理程序, 该程序主要实现了高速缓冲区缓存的功能。第二部分是文件系统底层程序, 主要实现了对 i 结点的分配与释放, 文件路径名与 i 结点的映射, 文件系统超级块的处理以及文件系统的选择处理功能。第三部分是文件数据的访问操作程序, 主要实现了对字符设备、管道、块设备、正规文件的数据访问功能。第四部分是文件系统的上层操作程序, 主要实现了文件的打开、关闭、创建等有关文件操作的系统调用。

Linux 的基本思想有两点: 第一, 一切都是文件; 第二, 每个软件都有确定的用途, 同时它们都尽可能被编写得更好。其中第一条详细来讲就是系统中的所有内容都归结为一个文件, 包括命令、硬件和软件设备、操作系统、进程等。

### 1.1.2 Linux 操作系统的特征

Linux 操作系统具有以下特征: 并发、共享、虚拟和异步性。

#### 1. 并发 (concurrency)

并发 (concurrency) 是指多个事件在同一时间段内发生。操作系统是一个并发系统, 包括各进程间的并发, 系统与应用间的并发。操作系统要完成这些并发过程的管理。并行 (parallel) 是指在同一时刻发生。在多道程序处理时, 宏观上并发, 微观上交替执行 (在单处理器情况下)。程序的静态实体是可执行文件, 而动态实体是进程 (或称做任务), 并发指的是进程。

#### 2. 共享 (sharing)

多个进程共享有限的计算机系统资源。操作系统要对系统资源进行合理分配和使用。资源在一个时间段内交替被多个进程所用。资源分配后到释放前, 不能被其他进程所用。

#### 3. 虚拟 (virtual)

虚拟 (virtual) 是指一个物理实体映射为若干个对应的逻辑实体——分时或分空间。虚拟是操作系统管理系统资源的重要手段, 可提高资源利用率。关于虚拟: CPU——每个用户 (进程) 的“虚处理器”; 存储器——每个进程都占有的地址空间 (指令+数据+堆栈); 显示设备——多窗口或虚拟终端 (virtual terminal)。





## 4. 异步性 (Asynchronism)

异步性 (Asynchronism) 也称不确定性, 指进程的执行顺序和执行时间的不确定性。关于异步性, 进程的运行速度不可预知: 分时系统中, 多个进程并发执行, “时走时停”, 不可预知每个进程的运行推进快慢。无论进程运行快慢, 结果应该相同——通过进程互斥和同步手段来保证。难以重现系统在某个时刻的状态 (包括重现运行中的错误)。性能保证: 实时系统与分时系统相似, 但通过资源预留以保证性能。

## 1.2 Linux 内核概述

Linux 内核是一个庞大而复杂的操作系统的核心, 不过尽管庞大, 但是却采用子系统和分层的形式, 很好地进行了组织。因此, 对于完整的 Linux 内核进行剖析和阅读也并非很难。通过阅读完整的 Linux 内核源代码, 将会真正地理解 Linux 操作系统的基本工作原理。

### 1.2.1 Linux 内核的版本

Linux 内核分为两种: 稳定版和开发版。稳定的内核具有工业级的强度, 可以广泛的在商业开发中应用。Linux 的稳定版内核大部分都是对一些 Bug 进行了修正或是在内核中加入一些新的驱动程序。而开发版的内核中许多内容变化得都比较快。而且由于开发者不断试验新的解决方案, 因此内核经常是不稳定的。

6

#### 1. Linux 版本的命名规则

Linux 通过命名机制来区分稳定的和处于开发中的内核。这种机制使用类似 num1.num2.num3, 三个用点分隔的数字代表不同版本的内核。其中 num1 是主版本号, num2 是次版本号, num3 是修订版本号。次版本号可以反映出该内核是一个稳定版本还是一个开发版本: 如果该数字是偶数, 那么此内核就是稳定版, 如果是奇数, 那么它就是开发版。例如, 版本号为 2.6.0 的内核, 它就是一个稳定版。这个内核的主版本号是 2, 次版本号是 6, 修订版本号是 0。头两个数字在一起可以描述内核系列。

#### 2. Linux 内核版本的升迁

Linux 内核发行版简史如图 1-2 所示。

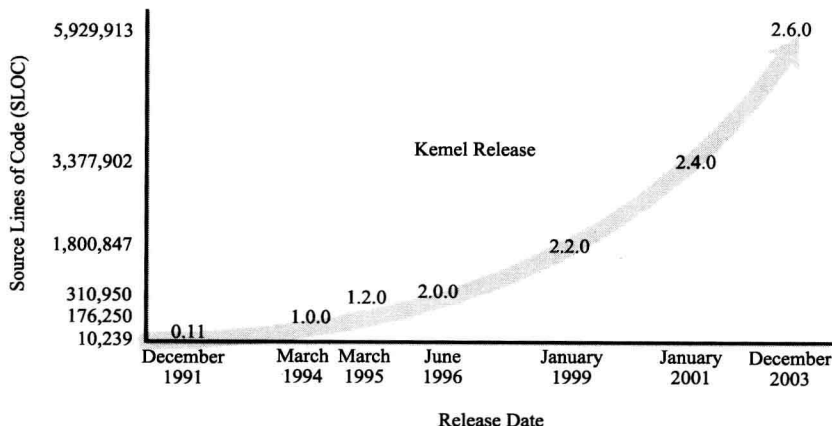


图 1-2 Linux 内核发行简史