

最详细的Linux简史来了——Linux大神带你领略它的前世今生



人民邮电出版社

已认证账号

154 人赞同了该文章

作者简介：陈华才，博士，高级工程师，《用芯探核 基于龙芯的Linux内核探索解析》的作者，负责龙芯3号处理器的内核开发。作为Linux内核中MIPS/ 龙芯3 号的Maintainer，向Linux内核官方提交了大量的代码。

1991年，芬兰大学生Linux Torvalds开发出最初的Linux，这个操作系统适用于基于Intel 80386微处理器的IBM PC兼容机。经过多年的发展，Linux已经可以在许多其他平台上运行，包括Alpha、Itanium (IA64)、MIPS、ARM、SPARC、MC680x0、PowerPC以及zSeries。

Linux最吸引人的一个优点就在于**它是一个自由的操作系统**：它的源代码基于GNU公共许可证（GNU Public License, GPL），是开放的，任何人都可以获得源码并研究它；只要得到源代码，就可以深入探究这个成功而又现代的操作系统。

Linux提倡自由、开源、共享、人人为我，我为人人。在GPL的号召下，全世界的Linux开发者组成了一个虚拟的开源社区。这是一种非常优秀的组织结构，尽管大家分布在世界各地，但是可以通过源代码和互联网进行高效的无障碍交流。大家既从开源社区获取资源，也把自己的贡献回馈给开源社区。

接下来，本文主要从**Linux内核的特点、Linux内核发展简史和Linux内核的开发模式**三个方面进行介绍，最后简单讲述什么是**长期维护稳定版**。

Linux内核的特点

从技术角度来说，**Linux只是操作系统内核，而不是一个完全类UNIX操作系统**，这是因为它不包含全部的UNIX应用程序，诸如文件系统实用程序、命令解释器、窗口系统、图形化桌面、系统管理员命令、文本编辑程序、编译开发程序等。以上这些应用程序大部分都可在GNU公共许可证下免费获得，因此包含Linux内核、基础运行环境（运行时库如GLibc）、编译环境（如GCC）、外壳程序（Shell，即命令解释器）和图形操作界面（GUI）的完整操作系统套件被称为**GNU/Linux**。尽管如此，在大多数情况下，仍用Linux来指代完整的GNU/Linux。



Linux 内核遵循IEEE POSIX标准（POSIX全称是Portable Operating System Interface of UNIX）。它包括了现代UNIX操作系统的全部特点，诸如虚拟存储、虚拟文件系统、内核线程、轻量级进程、UNIX信号量、SVR4进程间通信、支持内核抢占、对称多处理器（Symmetric Multi-Processor，SMP）系统等。

Linux内核是**一体化内核（或称宏内核）**操作系统，**宏内核的设计风格是“凡是可以在内核里实现的都在内核里实现”**。因此，除了异常/中断处理、内存管理和进程管理三大基本功能以外，文件系统、设备驱动和网络协议也放在内核层实现。

宏内核的优点是内核内部的各种互操作都可以通过函数调用实现，因此性能较好；而缺点是体积较大且理论上健壮性不太好（因为内部耦合性太高）。与宏内核相对的是**微内核**，最常见的是实现是GNU Hurd，其设计风格是**“凡是可以在内核里实现的都不在内核里实现”**，因此很多功能子系统被设计成了一种服务（进程）。

微内核的优点是体积较小且理论上更健壮（因内核本身的功能较少而不容易出错）；而缺点是操作系统的大量互操作都依赖于进程间通信（Inter-Process Communication，IPC），因此性能较差。微内核虽然把一些内核的核心功能剥离到了服务进程中，但重要的服务崩溃后实际上跟内核崩溃类似，因为整个系统也同样处于一个基本不可用的状态。

Linux虽然是宏内核，但是也吸收了一些微内核的优点，比如从1.0版本开始就可以通过模块化（将一些非核心的功能设计成可以运行时动态加载/卸载的内核模块）来减少内核核心部分的体积。

Linux内核发展简史

Linux内核从最初发布的0.01版直到4.4.x版，经历了“**史前时代**”“**奇偶时代**”“**快速演进时代**”和“**极速演进时代**”四个阶段（如图1所示）。

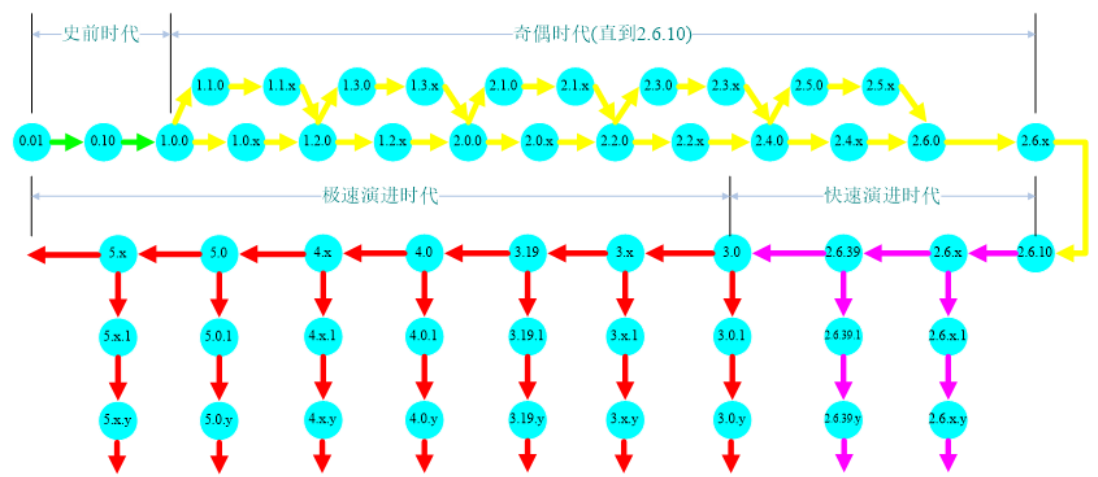


图1 Linux内核版本演进图

1. 史前时代（0.01~1.0.0）

版本更迭过程为：

0.01 → 0.02 → 0.10 → 0.11 → 0.12 → 0.95 → 0.96 → 0.97.x → 0.98.x → 0.99.x → 1.0.0

其中重要的版本有：

- 0.01：第一个版本；
- 0.02：第一个公开发布的版本；

- 0.11: 《Linux内核完全注释》使用的版本。



2. 奇偶时代 (1.0.0~2.6.10)

这个时期的版本号用a.b.c表示，其中a为主版本号，b为次版本号，c为修订号。

版本号变更的原则是：发生重大改变时升级主版本号，发生非重大改变时升级次版本号；次版本号为奇数表示开发版，次版本号为偶数表示稳定版；稳定版和开发版在修订号上各自升级演进，开发版达到稳定状态时，发布下一个稳定版。

比如1.0.x在尽量不引入新功能的前提下不断升级；同时1.1.x在不断开发新功能的状态下不断升级；当1.1.x的开发到足够稳定时，转变成1.2.x成为稳定版；同时新的开发版1.3.x诞生……

稳定版：1.0.x, 1.2.x, 2.0.x, 2.2.x, 2.4.x, 2.6.x

开发版：1.1.x, 1.3.x, 2.1.x, 2.3.x, 2.5.x

其中重要的版本有：

- 1.0.0: 第一个正式版本，支持模块化，开始支持网络；
- 1.2.0: 开始支持非X86架构；
- 2.0.0: 开始支持对称多处理（SMP）；
- 2.2.0: 开始被各种发行版大规模应用；
- 2.4.5: 开始有“中国制造”的代码（如LVS等）；
- 2.4.18: 《深入理解Linux内核（第2版）》所使用的版本；
- 2.6.0: 开始响彻天下：完全可抢占，O(1)调度器，SYSFS，X86_64支持，NUMA支持，NPTL支持……

3. 快速演进时代 (2.6.11~2.6.39)

从2.6.11开始，Linux内核界发生了两件大事：**第一件大事是抛弃了BitKeeper，转而开始用Git管理源代码；第二件大事是抛弃了奇偶版本法，转而使用a.b.c.d表示**，其中a为主版本号，b为次版本号，c为主修订号，d为次修订号。主修订号c的升级既包括新特性引入，也包括缺陷修订（Bugfix），次修订号d的升级只包括Bugfix。

这个阶段开发速度加快，版本号即便第三段相邻，差别也很大。在奇偶时代，2.4.5和2.4.6的差异不是很大；而在快速演进时代，2.6.36和2.6.37的差别会非常大，甚至堪比2.4.x和2.5.x之间的差异。

在演进如此迅速的时代，如果继续采用奇偶版本法会有什么问题？

首先，2.7版本开发持续时间会很长，不到2.8发布时，2.7加入的新特性无法得到利用。其次，2.7版本新特性同样很难后向移植（Backport）到2.6，因为代码差异太大。

这个阶段的重要版本有：

- 2.6.11.0: 《深入理解Linux内核》第三版所使用的版本；
- 2.6.20.0: 开始支持KVM虚拟化技术；
- 2.6.23.0: 开始支持龙芯2E，引入CFS调度器；
- 2.6.24.0: i386和x86_64合并成x86架构，缺省使用SLUB内存分配器；
- 2.6.33.0: 开始支持龙芯2F，在MIPS系列处理器上支持内核压缩；
- 2.6.38.0: 引入AutoGroup机制，大幅提升桌面应用体验，引入透明巨页（THP）。

4. 极速演进时代 (3.0~5.x)



在快速演进阶段，内核版本号的a.b一直保持为2.6没变，完全可以合二为一。与此同时，参与Linux内核开发的个人与单位越来越多，开始进入极速演进时代。**在这个阶段，版本号回归a.b.c表示法，a为主版本号，b为次版本号，c为修订号。**

在含义上，新的a相当于之前的a.b，新的b相当于之前的c，新的c相当于之前的d。次版本号b的升级既包括新特性引入，也包括缺陷修订（Bugfix），修订号c的升级只包括Bugfix。

关于每个版本的Linux内核都引入了什么新功能，可以参考官方的内核发行概述：

LinuxVersions - Linux Kernel Newbies

kernelnewbies.org/LinuxVersions

这个阶段的重要版本有：

- 3.6.0：开始支持龙芯1号；
- 3.8.0：引入调度实体负载跟踪机制（PELT），MIPS系列处理器开始支持透明巨页；
- 3.10.0：Radeon系列显卡开始支持高清视频解码（UVD）；
- 3.13.0：NUMA调度性能大幅度改进；
- 3.14.0：MIPS系列处理器开始支持FP64/O32；
- 3.15.0：开始支持龙芯3A1000，开始支持MIPS向量扩展（MSA）；
- 3.16.0：开始引入快速排队读写锁（qrwlock）。
- 3.17.0：开始支持龙芯3B1500，开始支持MIPS硬件页表遍历器（HTW）；
- 3.18.0：开始支持用GCC5编译内核；
- 4.0.0：开始支持在线补丁（LivePatching）和内核地址净化器（KASan）；
- 4.2.0：开始引入快速排队自旋锁（qspinlock），代码量达到2000万行；
- 4.5.0：MIPS开始支持IEEE754-2008标准，引入CGroup_V2；
- 4.7.0：开始龙芯3A2000，MIPS开始支持可变长ASID、48位虚拟地址空间、可重定位内核和内核地址空间布局随机化（KASLR），CPUFreq增加schedutil策略；
- 4.8.0：开始支持龙芯1C，完善支持软件MIPS KVM，内存页回收从基于管理区重构为基于NUMA节点；
- 4.9.0：MIPS开始引入通用内核，引入TCP拥塞控制算法BBR；
- 4.12.0：MIPS支持48位虚拟地址空间和硬件虚拟化（KVM/VZ），LivePatching使用每进程一致性模型（原来是全局一致性模型）；
- 4.13.0：开始支持龙芯3A3000，MIPS开始支持qspinlock/qrwlock；
- 4.15.0：开始支持RISC-V，X86引入KPTI（对付Meltdown漏洞）和Retpoline（对付Spectre漏洞）；
- 5.0.0：调度器引入EAS（节能感知）特征，块设备层全面切换到多队列模型（blk-mq），AMDGPU显卡驱动支持FreeSync，全面支持零拷贝网络。

Linux内核的开发模式

目前，Linux内核开发处于极速演进时代。在代码仓库管理上，有**主线仓库**（Mainline）、**稳定仓库**（Stable）、**未来仓库**（Linux-next）和**子系统仓库**（Subsystem，如Linux-mips）4大类，其关系如图2。

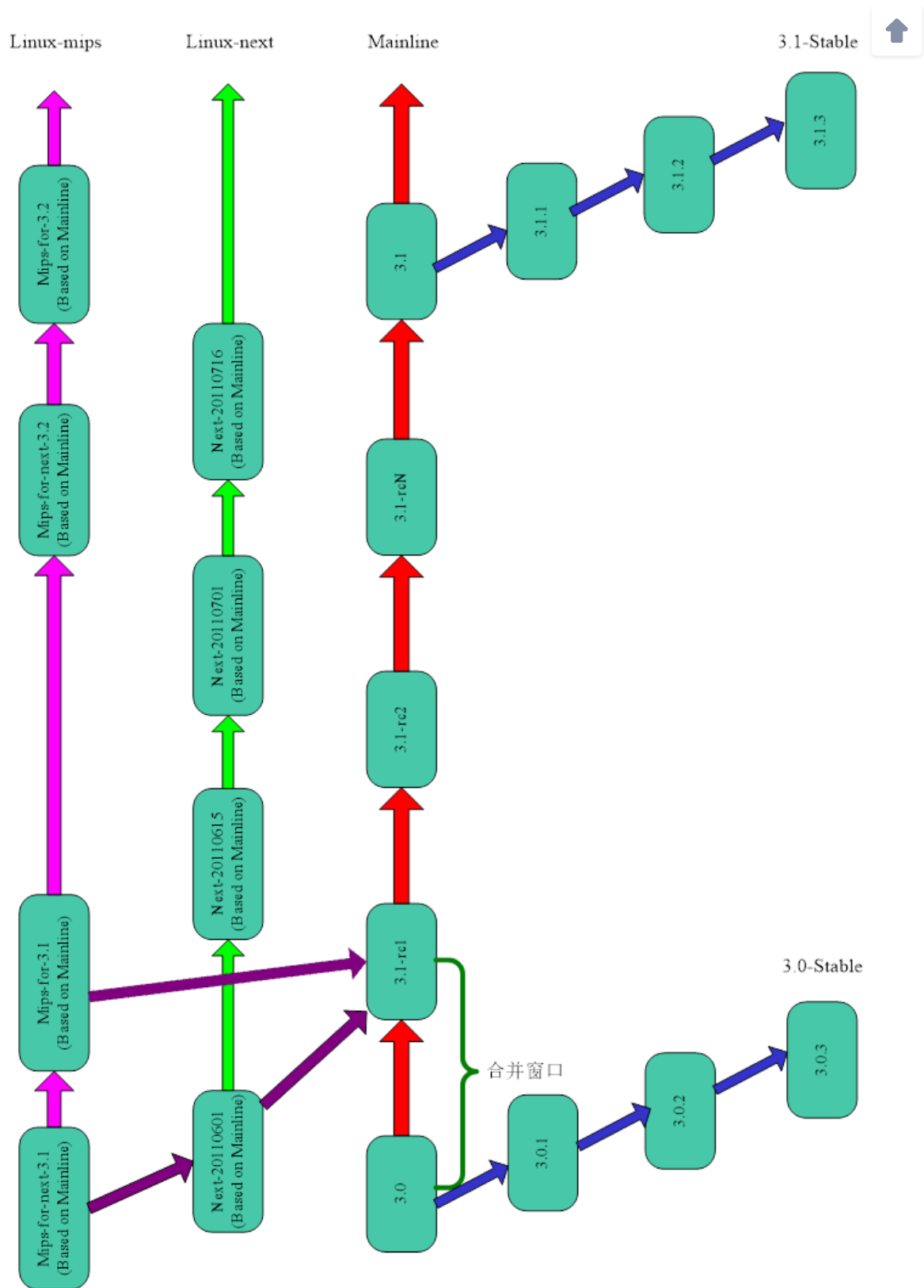


图2 Linux内核的四类代码仓库及其关系

绝大多数开发者所贡献的代码首先要接受子系统管理员（Maintainer）的审核，才能进入某个特定的子系统仓库；在进入子系统仓库以后，会通过未来仓库进行二次审核；二次审核通过以后，最后将进入主线仓库（偶尔也有跳过未来仓库，从子系统仓库直接进入主线仓库的情况）。可以说，代码进入子系统仓库以后才仅仅处于Alpha状态；进入未来仓库以后才算达到Beta状态；如果进入了主线内核，就相当于达到RC状态或者Final状态，算是被官方采纳了。通过这种多层次的严格审核，Linux内核的代码质量得到了极大的保证。

下面分别介绍这4类代码仓库。



1. 主线 (Mainline) 仓库

主线仓库是最重要的仓库，其升级规则是在次版本号上面升级演进，两个正式版之间会发布若干个候选版 (RC版)，如：

3.0 → 3.1-rc1 → 3.1-rcN → 3.1 → 3.2-rc1 →

某一个正式版和下一个候选版之间的时期叫做合并窗口期，比如3.0至3.1-rc1之间就是3.1的合并窗口。只有在合并窗口里面才允许增加新特性，其他阶段只允许缺陷修订 (Bugfix)。也就是说，如果开发者想让某个新特性进入到3.1内核，那么必须保证在3.1-rc1之前进入，否则就只能等待3.2的合并窗口了。

Maintainer (管理员)：Linus Torvalds

Git仓库地址：[git://git.kernel.org/pub/scm/...](https://git.kernel.org/pub/scm/...)

2. 稳定 (Stable) 仓库

稳定仓库基于主线仓库的正式版产生，在修订号上面升级演进，如3.0.x分支和3.1.x稳定分支仓库中的版本演进关系分别为：

3.0 → 3.0.1 → 3.0.2 → 3.0.3 → 3.0.N →

3.1 → 3.1.1 → 3.1.2 → 3.1.3 → 3.1.N →

稳定仓库的代码变更全都是缺陷修订 (Bugfix)，不引入新的特征。

Maintainer (管理员)：Greg Kroah-Hartman等

Git仓库地址：[git://git.kernel.org/pub/scm/...](https://git.kernel.org/pub/scm/...)

3. 未来 (Linux-next) 仓库

未来仓库的前身为Andrew Morton维护的Linux-mm，代码变更在进入下一版主线内核之前先到达这里。如果说主线仓库在功能上类似于奇偶时代的偶数版本 (稳定版) 的话，那么未来仓库在功能上类似于奇偶时代的奇数版本 (开发版)。未来仓库的版本命名规则是日期，如next-20151212。未来仓库会不定期合并主线仓库的代码，将其作为新一轮添加特性的基础 (Base) 代码。

Maintainer (管理员)：Stephen Rothwell

Git仓库地址：[git://git.kernel.org/pub/scm/...](https://git.kernel.org/pub/scm/...)

4. 子系统 (Subsystem) 仓库

子系统仓库为数众多，一般按体系结构 (arch)、驱动类型 (drivers) 进行分类。龙芯内核开发者比较关心的子系统仓库主要有以下两个。

- MIPS子系统：Maintainer为Ralf Baechle

Git仓库地址：[git://git.kernel.org/pub/scm/...](https://git.kernel.org/pub/scm/...)

- GPU子系统：Maintainer为David Airlie



和未来仓库一样，子系统仓库会不定期合并主线仓库的代码，将其作为新一轮添加特性的基础（Base）代码。

内核根目录的MAINTAINERS文件中列出所有现任的Maintainer及其相关信息，比如MIPS架构下面龙芯相关的三个子架构（即龙芯1号、龙芯2号和龙芯3号）的Maintainer信息如下。

MIPS/LOONGSON1 ARCHITECTURE

M: Keguang Zhang <keguang.zhang@gmail.com>
L: linux-mips@vger.kernel.org
S: Maintained
F: arch/mips/loongson32/
F: arch/mips/include/asm/mach-loongson32/
F: drivers/*/*loongson1*
F: drivers/*/*/*loongson1*

MIPS/LOONGSON2 ARCHITECTURE

M: Jiaxun Yang <jiaxun.yang@flygoat.com>
L: linux-mips@vger.kernel.org
S: Maintained
F: arch/mips/loongson64/fuloong-2e/
F: arch/mips/loongson64/lemote-2f/
F: arch/mips/include/asm/mach-loongson64/
F: drivers/*/*loongson2*
F: drivers/*/*/*loongson2*

MIPS/LOONGSON3 ARCHITECTURE

M: Huacai Chen <chenhc@lemote.com>
L: linux-mips@vger.kernel.org
S: Maintained
F: arch/mips/loongson64/
F: arch/mips/include/asm/mach-loongson64/
F: drivers/platform/mips/cpu_hwmon.c
F: drivers/*/*loongson3*
F: drivers/*/*/*loongson3*

MAINTAINERS文件中包含一个个子系统的条目。

每个条目的开头第一行是关于子系统的描述，比如，龙芯1号的MIPS/LOONGSON1 ARCHITECTURE，龙芯2号的MIPS/ LOONGSON2 ARCHITECTURE和龙芯3号的MIPS/LOONGSON3 ARCHITECTURE。

M开头的行是Maintainer的姓名和电子邮箱，龙芯1号的Maintainer是张科广，龙芯2号的Maintainer是杨嘉勋，这两位都是开源社区的龙芯爱好者，而龙芯3号的Maintainer就是《用“芯”探核：基于龙芯的Linux内核探索解析》的作者。L开头的行是该子系统用于开发交流

的邮件列表；S开头的行是该子系统的维护状态；F开头的行是该子系统所涉及的主要源代码文件的路径。



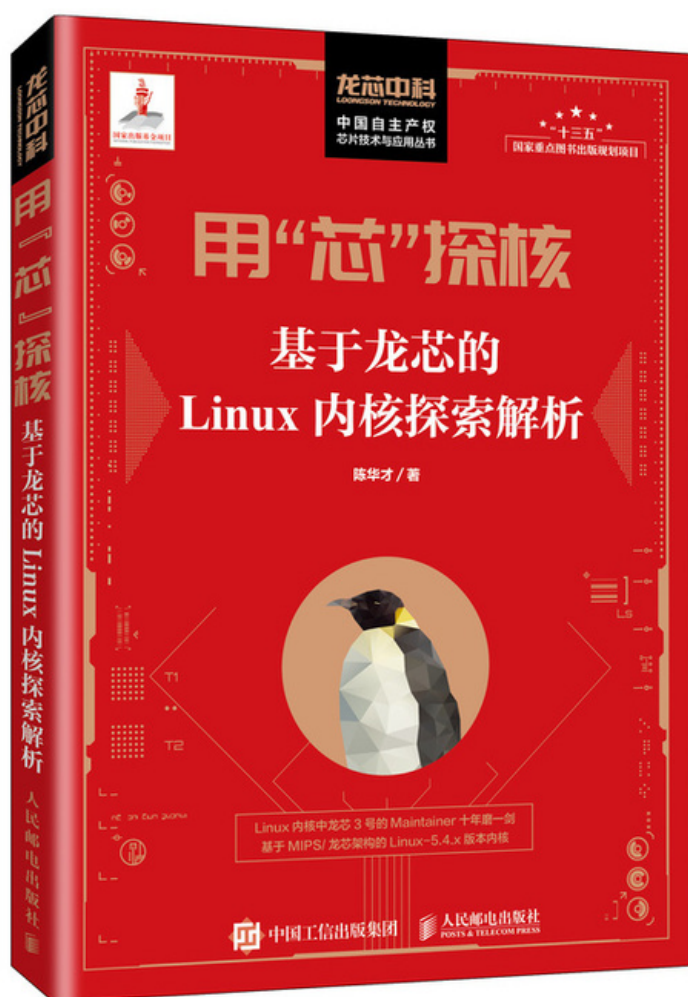
长期维护稳定版

所谓的长期维护稳定版（Long Term Stable, LTS），实际上是一种特殊的稳定版本（Stable）。Stable版本的缺陷修订（即Bugfix）实际上是Mainline版本中缺陷修订的后向移植（即Backport）。普通的Stable版本的维护时间为3个月左右，因此当Mainline仓库中下一版的正式版发布，上一版的Stable分支就不再继续升级（EOL, End Of Life）。而LTS版本的维护时间为2年左右，也可能更长。同时维护的LTS版本为5个左右，当一个新的LTS版本被选中时，一般最老的LTS就不再继续升级（EOL）。

Linux内核官方选择长期维护稳定版本的依据大致有几点：具有里程碑意义（如3.0），是过去一段时间内引入的新特性的集大成者，或者是被Redhat、Debian等著名发行版采用的版本。曾经被选为长期维护版本的内核有：2.6.16.x、2.6.27.x、2.6.32.x、2.6.33.x、2.6.34.x、2.6.35.x、3.0.x、3.2.x、3.4.x、3.10.x、3.12.x、3.14.x、3.16.x、3.18.x、4.1.x、4.4.x、4.9.x、4.14.x、4.19.x和5.4.x。

相关阅读

《用“芯”探核 基于龙芯的Linux内核探索解析》





用“芯”探核 基于龙芯的Linux内核探索解析

京东

¥116.80

[去购买 >](#)

这是一本基于龙芯平台，结合Linux-5.4.x 版本的内核源代码来解析Linux 内核的书籍。

本书首先介绍了有关龙芯处理器和Linux 内核的基础知识，然后重点讲解了内核启动、异常与中断、内存管理、进程管理、显卡驱动、网卡驱动和电源管理这7 大板块的内容。本书甚少涉及代码的细枝末节，而是重点关注代码实现的主干流程，并且创造性地引入了树形视图和链式视图这两种比流程图更好用的代码解析方法。

本书适合Linux 系统相关的开发人员，特别是基于龙芯处理器做内核开发的技术人员学习参考。

=====

赠人玫瑰，手留余香，不要忘记点赞、收藏、关注 [@人民邮电出版社](#) 哦~

一键三连，感恩有你~

编辑于 2020-08-27 18:47

[Linux](#)

[Linux 内核](#)

[Linux 运维](#)

写下你的评论...

评论内容由作者筛选后展示



还没有评论，发表第一个评论吧

文章被以下专栏收录



Linux内核

共同探讨Linux内核的奥秘



推荐阅读



一些经典 Linux 命令的现代替代品 | Linux 中国

Linux... 发表于Linux...



在 Linux 上调整命令历史 | Linux 中国

Linux... 发表于Linux...



用户必会的 20 个 Linux 基础命令 | Linux 中国

Linux... 发表于Lin