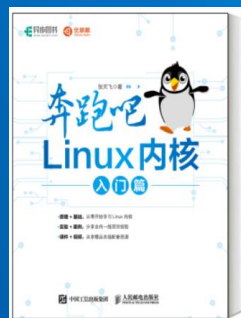




# 奔跑吧Linux内核\*入门篇

## 第十二章 参与开源社区

笨叔叔



# 目录

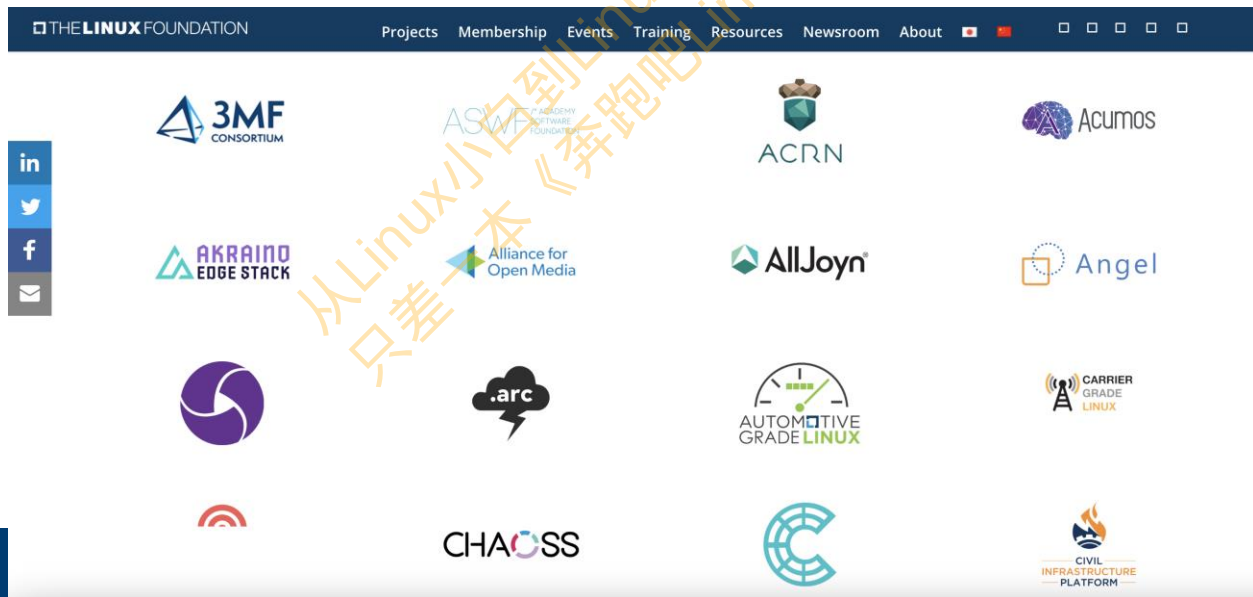
- 开源社区
- 开源许可证
- Linux基金会
- Linux内核社区
- 如何参与开源社区
- 实验

# 开源社区

- 开源社区又称开放源代码社区，一般由拥有共同兴趣爱好的人所组成，根据相应的开源软件许可证协议公布软件源代码的网络平台，同时也为网络成员提供一个自由学习交流的空间
- 开源软件（Free Software）从20世纪80年代诞生以来，就像星星之火，今天已经成为软件开发行业的中坚力量
- 1983年，理查德（Richard Stallman）发起了GNU（GUN's Not UNIX）项目
- 1991年10月5日，Linus Torvalds在comp.os.minix新闻组上发布新闻，正式对外宣告Linux内核诞生

# Linux基金会

- 2000年，由开源软件发展实验室（Open Source Development Labs）和自由标准组织（Free Standards Group）联合成立Linux基金会
- 除了托管Linux内核项目，还有托管差不多100个开源项目



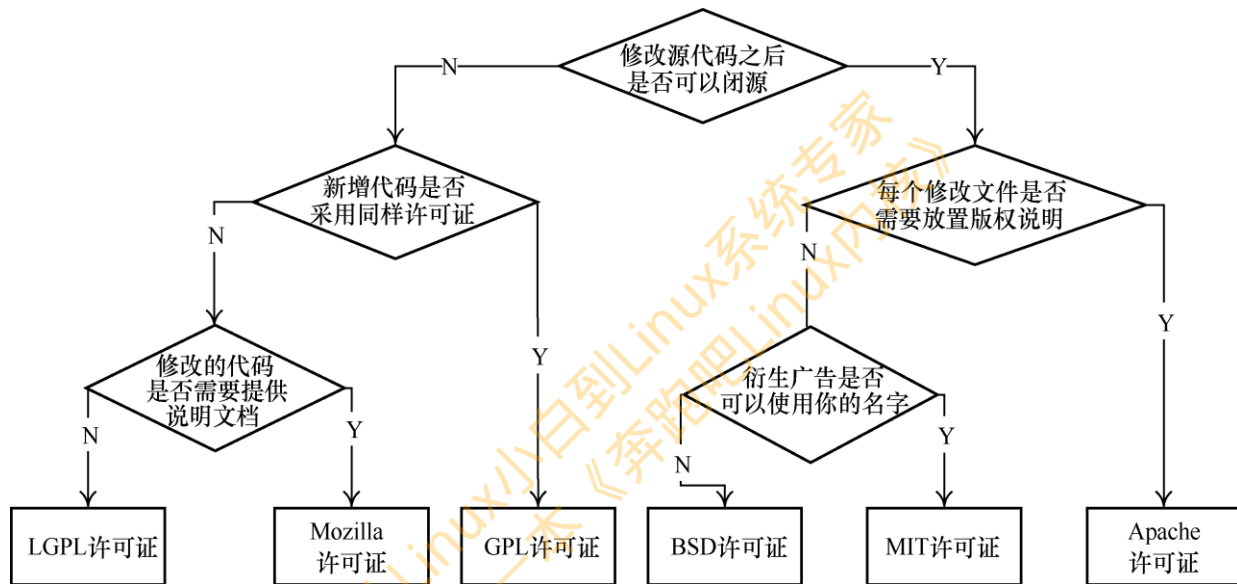
# 开源协议

## ➤ GPL许可证

- ✓ 1989年发布GPL v1
  - 发布可执行的二进制代码，就必须同时发布源代码
- ✓ 1991年发布GPL v2
  - 当采用了部分的GPL协议相关的软件，那么这个软件整体就必须采用GPL协议
- ✓ 2005年发布GPL v3
  - 任何公司或者实体以GPL v3协议发布软件，那么它将永远以GPL v3协议发布，并且原专利拥有者在任何时候不具备收取专利费的权力。
  - 专利报复条款。禁止发布软件的公司或者实体向被许可人发起专利诉讼。
  - Tivo化。

## ➤ BSD许可证

## ➤ Apache许可证



# Linux内核社区

- Linus是导演
- 采用GPL v2许可证
- 超过百家顶尖公司参与，数千名开发人员
- 采用邮件列表方式进行技术讨论
- 截止2017年，Linux内核源代码已经超过2000万行代码

# Linux内核开发模式

- Linux内核的开发模式完全由社区来主导。
  - ✓ Linux是Linux内核的最大的维护者和导演
  - ✓ 每个子模块都有各自维护者
  - ✓ 在邮件列表为讨论主战场，比如LKML

Kernel Version	Release Date	Days of Development
3.19	2015-02-08	63
4.0	2015-04-12	63
4.1	2015-06-21	70
4.2	2015-08-30	70
4.3	2015-11-01	63

Kernel Version	Release Date	Days of Development
4.4	2016-01-10	70
4.5	2016-03-13	63
4.6	2016-05-15	63
4.7	2016-07-24	70

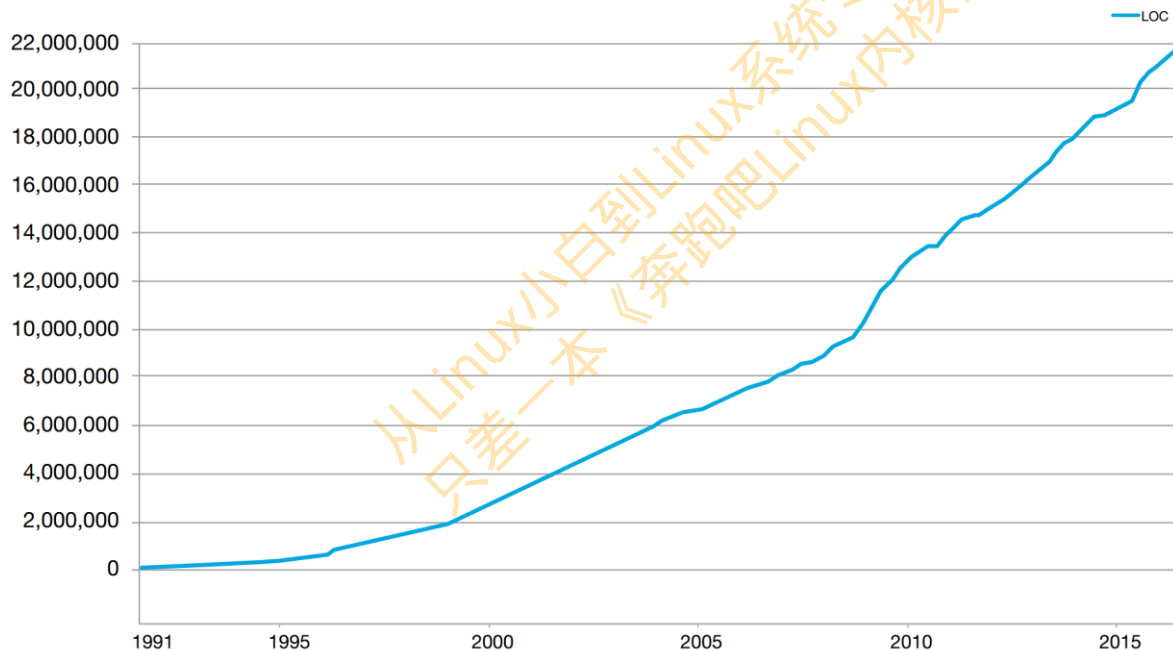
大约60-70天一个新内核版本发布



# Linux内核行数的变化

- Linux内核从1991年的1w行代码发展到超过200w行代码

Total Lines of Code in the Linux Kernel



# Linux内核开发参与者

- Linux内核参与的开发者分布全球，数以千计的开发者，几个顶尖的IT公司

Kernel Version	Developers	Companies
3.19	1,451	230
4.0	1,458	214
4.1	1,539	238
4.2	1,591	251
4.3	1,625	211

Kernel Version	Developers	Companies
4.4	1,575	220
4.5	1,537	231
4.6	1,678	243
4.7	1,582	221

# 主要的内核开发贡献者

Those developers are:

Name	Changes	Percent
H Hartley Sweeten	5,960	1.1%
Al Viro	5,433	1.0%
Takashi Iwai	4,723	0.8%
Mark Brown	3,960	0.7%
David S. Miller	3,950	0.7%
Mauro Carvalho Chehab	3,943	0.7%
Tejun Heo	3,852	0.7%
Johannes Berg	3,707	0.7%
Russell King	3,467	0.6%
Thomas Gleixner	3,233	0.6%
Hans Verkuil	3,119	0.6%
Greg Kroah-Hartman	3,117	0.6%
Ingo Molnar	2,873	0.5%
Joe Perches	2,778	0.5%
Christoph Hellwig	2,697	0.5%

Name	Changes	Percent
Eric Dumazet	2,633	0.5%
Axel Lin	2,604	0.5%
Dan Carpenter	2,562	0.5%
Geert Uytterhoeven	2,460	0.4%
Laurent Pinchart	2,381	0.4%
Alex Deucher	2,340	0.4%
Bartlomiej Zolnierkiewicz	2,279	0.4%
Trond Myklebust	2,269	0.4%
Paul Mundt	2,268	0.4%
Daniel Vetter	2,224	0.4%
Ben Skeggs	2,216	0.4%
Arnd Bergmann	2,199	0.4%
Lars-Peter Clausen	2,176	0.4%
Arnaldo Carvalho de Melo	2,107	0.4%
Ralf Baechele	2,097	0.4%

# 主要的参与公司

The most active companies over the 3.19 to 4.7 development cycles were:

Company	Changes	Percent
Intel	14,384	12.9%
Red Hat	8,987	8.0%
none	8,571	7.7%
unknown	7,582	6.8%
Linaro	4,515	4.0%
Samsung	4,338	3.9%
SUSE	3,619	3.2%
IBM	2,995	2.7%
consultants	2,938	2.6%
Renesas Electronics	2,239	2.0%
Google	2,203	2.0%
AMD	2,100	1.9%
Texas Instruments	1,917	1.7%
ARM	1,617	1.4%
Oracle	1,528	1.4%

Company	Changes	Percent
Outreachy	1,524	1.4%
Vision Engraving Systems	1,456	1.3%
Free Electrons	1,453	1.3%
NXP Semiconductors	1,445	1.3%
Mellanox	1,404	1.3%
Atmel	1,362	1.2%
Broadcom	1,237	1.1%
NVidia	1,146	1.0%
Code Aurora Forum	1,033	0.9%
Imagination Technologies	963	0.9%
Huawei Technologies	937	0.8%
Facebook	877	0.8%
Pengutronix	790	0.7%
Cisco	692	0.6%
Qualcomm	656	0.6%

# 参与开源社区

- 企业参与开源社区的好处：
  - ✓ 获取竞争优势以及提升品牌形象
  - ✓ 降低开发成本
  - ✓ 提升代码质量
- 个人参与开源社区的好处：
  - ✓ 提升综合开发能力。
  - ✓ 提高英语能力。
  - ✓ 激发工作激情。
  - ✓ 获取更多的工作机会

从Linux小白到Linux系统专家  
只差一本《奔跑吧Linux内核》

# 如何参与开源项目

## ➤ 订阅邮件列表

### Majordomo lists at VGER.KERNEL.ORG

REMEMBER: Subscription to these lists go via <majordomo@vger.kernel.org> !

Note about archives: Listed archives are those that have been reported to vger's maintainers, or that we have found out otherwise. As things are, *list of archives is not complete.*

[alsa-devel](#), [autofs](#), [backports](#), [ceph-devel](#), [cgroups](#), [cpufreq](#), [dash](#), [dccp](#), [devicetree-compiler](#), [devicetree-spec](#), [devicetree](#), [dmaengine](#), [dwarves](#), [ecryptfs](#), [fio](#), [fstests](#), [git-commits-24](#), [git-commits-head](#), [git](#), [hail-devel](#), [initramfs](#), [irda-users](#), [kernel-janitors](#), [kernel-packagers](#), [kernel-testers](#), [keyrings](#), [kvm-commits](#), [kvm-ia64](#), [kvm-ppc](#), [kvm](#), [lartc](#), [libzbc](#), [linux-8086](#), [linux-acpi](#), [linux-admin](#), [linux-alpha](#), [linux-apl](#), [linux-apps](#), [linux-arch](#), [linux-arm-msm](#), [linux-assembly](#), [linux-bbs](#), [linux-bcache](#), [linux-block](#), [linux-bluetooth](#), [linux-btrfs](#), [linux-c-programming](#), [linux-can](#), [linux-cifs](#), [linux-clk](#), [linux-config](#), [linux-console](#), [linux-covertity](#), [linux-crypto](#), [linux-diald](#), [linux-doc](#), [linux-edac](#), [linux-efi](#), [linux-embedded](#), [linux-ext4](#), [linux-fbdev](#), [linux-fido](#), [linux-fpga](#), [linux-fscrypt](#), [linux-fs-devel](#), [linux-fsf](#), [linux-ftio](#), [linux-gcc](#), [linux-gpio](#), [linux-hams](#), [linux-hexagon](#), [linux-hotplug](#), [linux-hwmon](#), [linux-i2c](#), [linux-ia64](#), [linux-ibcs2](#), [linux-ide](#), [linux-iiio](#), [linux-input](#), [linux-integrity](#), [linux-ipx](#), [linux-isd9](#), [linux-japanese](#), [linux-kbuild](#), [linux-kernel-announce](#), [linux-kernel-announce.posters](#), [linux-kernel](#), [linux-kselftest](#), [linux-laptop](#), [linux-leds](#), [linux-linuxss](#), [linux-lugnuts](#), [linux-m68k-cvscommit](#), [linux-m68k](#), [linux-man](#), [linux-mca](#), [linux-media](#), [linux-metag](#), [linux-mmc](#), [linux-modules](#), [linux-msdos-devel](#), [linux-msdos](#), [linux-new-lists](#), [linux-newbie](#), [linux-next](#), [linux-nfs](#), [linux-niifs](#), [linux-numa](#), [linux-omap](#), [linux-opengl](#), [linux-parisc](#), [linux-pci](#), [linux-perf-users](#), [linux-pm](#), [linux-ppp](#), [linux-pwm](#), [linux-raid](#), [linux-rdma](#), [linux-remoteproc](#), [linux-renesas-soc](#), [linux-rt-users](#), [linux-rtc](#), [linux-s390](#), [linux-samsung-soc](#), [linux-scsi](#), [linux-sctp](#), [linux-security-module](#), [linux-serial](#), [linux-sh](#), [linux-smp](#), [linux-soc](#), [linux-sound](#), [linux-sparse](#), [linux-spi](#), [linux-standards](#), [linux-svgalib](#), [linux-tape](#), [linux-tegra](#), [linux-tftp-commits](#), [linux-trace-devel](#), [linux-trace-users](#), [linux-unionsfs](#), [linux-usb](#), [linux-userfs](#), [linux-watchdog](#), [linux-wireless](#), [linux-word](#), [linux-wpan](#), [linux-x11](#), [linux-x25](#), [linux-x86-64](#), [linux-xfs](#), [live-patching](#), [lpc-netdev](#), [lvs-devel](#), [mim-commits](#), [netdev](#), [netfilter-devel](#), [netfilter](#), [perlbook](#), [platform-driver-x86](#), [reiserfs-devel](#), [smatch](#), [sparclinux](#), [stable-commits](#), [stable-rt](#), [stable](#), [stgt](#), [target-devel](#), [trinity](#), [ultralinux](#), [util-linux](#), [xdp-newbies](#).

## Linux内核支持的邮件列表

## ➤ 加入IRC频道

# 如何参与开源项目

- 关注缺陷管理系统
- 参与完善文档
- 参与开源活动

从Linux小白到Linux系统专家  
只差一本《奔跑吧Linux内核》

# 实验

从Linux小白到Linux系统专家  
只差一本《奔跑吧Linux内核》



# 实验1：使用cppcheck检查代码

## ➤ 实验目的

- ✓ 学会使用代码缺陷静态检测工具完善代码质量。

## ➤ 实验步骤

- ✓ 1) cppcheck是一个C/C++的代码缺陷静态检测工具。它不仅可以检测代码中的语法错误，还可以检测出编译器检查不出来的缺陷类型，从而帮助程序员提升代码质量。

从Linux小白到Linux系统专家  
只差一本《奔跑吧Linux内核》

# 实验2：提交第一个Linux内核补丁

## ➤ 实验目的

- ✓ 熟悉在Linux内核社区提交补丁的基本流程。

## ➤ 实验步骤

- ✓ 给Linux内核社区提交第一个补丁会涉及几个方面的问题：
  - 一是如何发现内核的缺陷
  - 二是如何制作补丁
  - 三是如何发送补丁。

从Linux小白到Linux系统专家  
只差一本《奔跑吧Linux内核》

# 实验3：管理和提交多个补丁组成的补丁集

## ➤ 实验目的

- ✓ 学会如何管理和提交多个补丁组成的补丁集。

## ➤ 实验步骤

- ✓ 如果读者订阅了Linux社区的邮件列表，就会发现有一些补丁集有几个甚至几十个补丁，而且会不断地发送新的版本。
- ✓ 读者通常会有如下的疑问。
  - ✓ 这些补丁集是如何生成的呢？
  - ✓ 当制作新版本的补丁集时，如何基于最新的Linux分支上进行？
  - ✓ 面对庞大的补丁集，如果社区针对某几个补丁有修改意见，那该如何制作新版的补丁集？

主题	通信者	日期
[PATCH v9 00/24] Speculative page faults	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 01/24] mm: Introduce CONFIG_SPECULATIVE_PAGE_FAULT	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 02/24] x86/mm: Define CONFIG_SPECULATIVE_PAGE_FAULT	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 03/24] powerpc/mm: Define CONFIG_SPECULATIVE_PAGE_FAULT	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 04/24] mm: Prepare for FAULT_FLAG_SPECULATIVE	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 05/24] mm: Introduce pte_spinlock for FAULT_FLAG_SPECULATIVE	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 06/24] mm: make pte_unmap_same compatible with SPF	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 07/24] mm: VMA sequence count	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 08/24] mm: Protect VMA modifications using VMA sequence count	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 10/24] mm: Protect SPF handler against anon_vma changes	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 11/24] mm: Cache some VMA fields in the vm_fault structure	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 12/24] mm/migrate: Pass vm_fault pointer to migrate_misplaced_page()	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 13/24] mm: Introduce __lru_cache_add_active_or_unevictable	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 14/24] mm: Introduce __maybe_mkwrite()	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 15/24] mm: Introduce __vm_normal_page()	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 16/24] mm: Introduce __page_add_new_anon_rmap()	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 17/24] mm: Protect mm_rb tree with a rwlock	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 18/24] mm: Provide speculative fault infrastructure	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 19/24] mm: Adding speculative page fault failure trace events	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 20/24] perf: Add a speculative page fault sw event	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 21/24] perf tools: Add support for the SPF perf event	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 22/24] mm: Speculative page fault handler return VMA	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 23/24] x86/mm: Add speculative pagefault handling	Laurent Dufour	3/14/2018 1:59 AM
[PATCH v9 24/24] powerpc/mm: Add speculative page fault	Laurent Dufour	3/14/2018 1:59 AM

## 多个patch组成的补丁集示例：Speculative Page Faults补丁集

# 实验4：在Gitee中创建一个开源项目

## ➤ 实验目的

- ✓ 熟悉如何创建和提交一个开源项目。

## ➤ 实验步骤

从Linux小白到Linux系统专家  
只差一本《奔跑吧Linux内核》

# BACKUP

从Linux小白到Linux系统专家  
只差一本《奔跑吧Linux内核》

shop115683645.taobao.com

## Linux视频课程



微信公众号：奔跑吧 linux 社区

1. > 一键订阅，持续更新
2. > 最有深度和广度的 Linux 视频
3. > 手把手解读 Linux 内核代码
4. > 紧跟 Linux 开源社区技术热点
5. > 笨叔叔的 VIP 私密群答疑
6. > 图书 + 视频，全新学习模式

shop115683645.taobao.com

配套视频 **旗舰篇**

第**1**季  
内存管理



规划中



旗舰篇一次订阅，持续更新

- |     |                      |
|-----|----------------------|
| 第二季 | 进程管理和调度 / 中断 / 锁（已出） |
| 第三季 | 虚拟化                  |
| 第四季 | Linux 内核和应用开发调试必杀技   |
| 第五季 | 红帽系列                 |



# 第1季旗舰篇课程目录

课程名称	时长
序言一：Linux内核学习方法论	0:09:13
序言二：学习前准备	
序言2.1 Linux发行版和开发板的选择	0:13:56
序言2.2 搭建Qemu+gdb单步调试内核	0:13:51
序言2.3 搭建Eclipse图形化调试内核	0:10:59
实战运维1：查看系统内存信息的工具（一）	0:20:19
实战运维2：查看系统内存信息的工具（二）	0:16:32
实战运维3：读懂内核log中的内存管理信息	0:25:35
实战运维4：读懂 proc meminfo	0:27:59
实战运维5：Linux运维能力进阶线路图	0:09:40
实战运维6：Linux内存管理参数调优（一）	0:19:46
实战运维7：Linux内存管理参数调优（二）	0:31:20
实战运维8：Linux内存管理参数调优（三）	0:22:58
运维高级如何单步调试RHEL—CENTOS7的内核一	0:15:45
运维高级如何单步调试RHEL—CENTOS7的内核二	0:41:28
vim:打造比source insight更强更好用的IDE（一）	0:24:58
vim:打造比source insight更强更好用的IDE（二）	0:20:28
vim:打造比source insight更强更好用的IDE（三）	0:23:25
实战git项目和社区patch管理	
2.0 Linux内存管理背景知识介绍	
奔跑2.0.0 内存管理硬件知识	0:15:25
奔跑2.0.1 内存管理总览一	0:23:27
奔跑2.0.2 内存管理总览二	0:07:35
奔跑2.0.3 内存管理常用术语	0:09:49
奔跑2.0.4 内存管理究竟管些什么东西	0:28:02
奔跑2.0.5 内存管理代码框架导读	0:38:09
2.1 Linux内存初始化	
奔跑2.1.0 DDR简介	0:06:47
奔跑2.1.1 物理内存三大数据结构	0:19:39
奔跑2.1.2 物理内存初始化	0:11:13
奔跑2.1 内存初始化之代码导读一	0:43:54
奔跑2.1 内存初始化之代码导读二	0:23:31

奔跑2.1 代码导读C语言部分（二）	0:21:28
2.2 页表的映射过程	
奔跑2.2.0 ARM32页表的映射	0:08:54
奔跑2.2.1 ARM64页表的映射	0:10:58
奔跑2.2.2 页表映射例子分析	0:11:59
奔跑2.2.3 ARM32页表映射那些奇葩的事	0:09:42
2.3 内存布局图	
奔跑2.3.1 内存布局一	0:10:35
奔跑2.3.2 内存布局二	0:13:30
2.4 分配物理页面	
奔跑2.4.1 伙伴系统原理	0:10:10
奔跑2.4.2 Linux内核中的伙伴系统和碎片化	0:11:14
奔跑2.4.3 Linux的页面分配器	0:21:37
2.5 slab分配器	
奔跑2.5.1 slab原理和核心数据结构	0:18:36
奔跑2.5.2 Linux内核中slab机制的实现	0:16:56
2.6 vmalloc分配	
奔跑2.6 vmalloc分配	0:15:48
2.7 VMA操作	
奔跑2.7 VMA操作	0:16:42
2.8 malloc分配器	
奔跑2.8.1 malloc的三个迷惑	0:17:41
奔跑2.8.2 内存管理的三个重要的函数	0:17:38
2.9 mmap分析	
奔跑2.9 mmap分析	0:23:14
2.10 缺页中断处理	
奔跑2.10.1 缺页中断一	0:31:07
奔跑2.10.2 缺页中断二	0:16:58
2.11 page数据结构	
奔跑2.11 page数据结构	0:29:41
2.12 反向映射机制	
奔跑2.12.1 反向映射机制的背景介绍	0:19:01
奔跑2.12.2 RMAP四部曲	0:07:31
奔跑2.12.3 手撕Linux2.6.11上的反向映射机制	0:07:35
奔跑2.12.4 手撕Linux4.x上的反向映射机制	0:10:08
2.13 回收页面	
奔跑2.13 页面回收一	0:16:07
奔跑2.13 页面回收二	0:11:41

奔跑2.11 page数据结构	0:25:41
2.12 反向映射机制	
奔跑2.12.1 反向映射机制的背景介绍	0:19:01
奔跑2.12.2 RMAP四部曲	0:07:31
奔跑2.12.3 手撕Linux2.6.11上的反向映射机制	0:07:35
奔跑2.12.4 手撕Linux4.x上的反向映射机制	0:10:08
2.13 回收页面	
奔跑2.13 页面回收一	0:16:07
奔跑2.13 页面回收二	0:11:41
2.14 匿名页面的生命周期	0:26:16
2.15 页面迁移	0:19:07
2.16 内存规整	0:24:03
2.17 KSM	0:28:17
2.20 Meltdown漏洞分析	
奔跑2.20.1 Meltdown背景知识	0:10:13
奔跑2.20.2 CPU体系结构之指令执行	0:11:25
奔跑2.20.3 CPU体系结构之乱序执行	0:11:03
奔跑2.20.4 CPU体系结构之异常处理	0:03:48
奔跑2.20.5 CPU体系结构之cache	0:10:56
奔跑2.20.6 进程地址空间和页表及TLB	0:17:39
奔跑2.20.7 Meltdown漏洞分析	0:06:04
奔跑2.20.8 Meltdown漏洞分析之x86篇	0:12:07
奔跑2.20.9 ARM64上的KPTI解决方案	0:25:39
代码导读	
奔跑2.1 内存初始化之代码导读一	0:43:54
奔跑2.1 内存初始化之代码导读二	0:23:31
奔跑2.1 代码导读C语言部分（一）	0:27:34
奔跑2.1 代码导读C语言部分（二）	0:21:28
代码导读3页表映射	1:12:40
代码导读4分配物理页面	0:55:57
git入门和实战	
git入门与实战：节目总览	0:08:48
git入门与实战1：建立本地的git仓库	0:30:53
git入门与实战2：快速入门	0:12:45
git入门与实战3：分支管理	0:24:27
git入门与实战4：冲突解决	0:20:20
git入门和实战5：提交更改	0:12:15
git入门和实战6：远程版本库	0:13:26
git入门和实战7：内核开发和实战	0:15:52
git入门和实战8：实战rebase到最新Linux内核代码	0:18:07
git入门和实战9：给内核发补丁	0:13:57

## 第2季旗舰篇课程目录

课程名称	时长
进程管理	
进程管理1基本概念	0:52:16
进程管理2进程创建	0:53:24
进程管理3进程调度	0:54:51
进程管理4多核调度	0:49:38
中断管理	
中断管理1基本概念	1:04:27
中断管理2中断处理part1	0:46:28
中断管理2中断处理part2	0:10:19
中断管理3下半部机制	0:55:57
中断管理4面试题目	1:13:57
锁机制	
锁机制入门1基本概念	0:56:16
锁机制入门2-Linux常用的锁	0:54:01



实战死机专题课程目录	
课程名称	时长
<b>上集x86_64</b>	
实战死机专题（上集）part1-kdump+crash介绍	0:30:09
实战死机专题（上集）part2-crash命令详解	0:28:15
实战死机专题（上集）part3-实战lab1	0:12:38
实战死机专题（上集）part4-实战lab2	0:11:03
实战死机专题（上集）part4-实战lab3	0:06:48
实战死机专题（上集）part4-实战lab4	0:15:28
实战死机专题（上集）part4-实战lab5	0:12:21
实战死机专题（上集）part4-实战lab6	0:24:07
实战死机专题（上集）part4-实战lab7	0:59:34
<b>下集arm64</b>	
实战死机专题(下集)part1	0:13:19
实战死机专题(下集)part2	0:20:47
实战死机专题(下集)part3	0:11:22
实战死机专题(下集)part4	0:33:01

全程约5小时高清，140多页ppt，8大实验，基于x86\_64的**Centos 7.6**和**arm64**，提供全套实验素材和环境。全面介绍kdump+crash在死机黑屏方面的实战应用，全部案例源自线上云服务器和嵌入式产品开发实际案例！



扫码识别

微店二维码



淘宝店二维码



微信号: Running-LinuxKernel

《奔跑吧Linux内核 \* 入门篇》相关的免费视频，或者更多更精彩更in的内容，请关注奔跑吧Linux社区微信公众号



# 奔跑吧 LINUX社区



旗舰篇一次订阅，持续更新

微信号: Runing-LinuxKernel