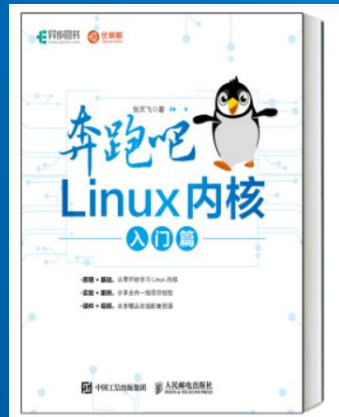




奔跑吧Linux内核*入门篇

第一章 LINUX系统入门

笨叔叔



目 录

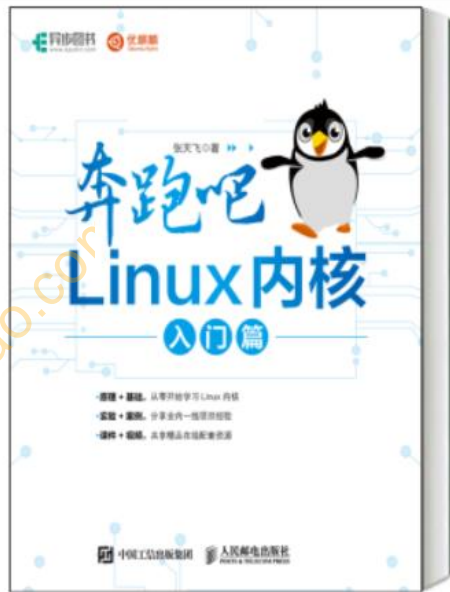
- 入门篇简介和导读
- Linux系统简介
- Linux内核设计之道
- Linux内核开发模式
- Linux内核架构Overview
- Linux内核学习方法

入门篇简介和导读



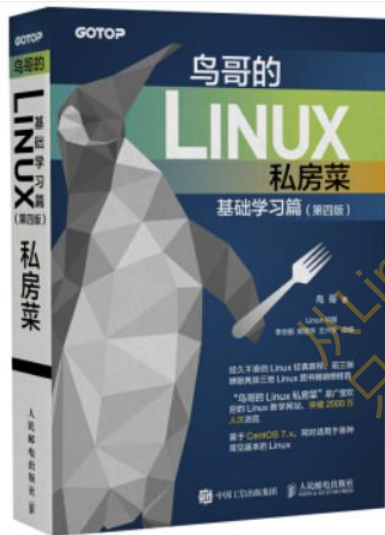
《奔跑吧Linux内核》入门篇特色

- ✓ 零基础学习Linux内核
- ✓ 学习最新开发工具Vim8, Git等
- ✓ 内容循序渐进, 深入浅出学习内核核心技术
- ✓ 70多个创新实验, 突出动手能力
- ✓ 融入最新开源社区开发理念
- ✓ 参与社区开发
- ✓ 提供精美课件
- ✓ 提供实验参考代码
- ✓ 提供全套实验环境 – vmware+优麒麟Linux镜像
- ✓ 提供免费补充高清视频

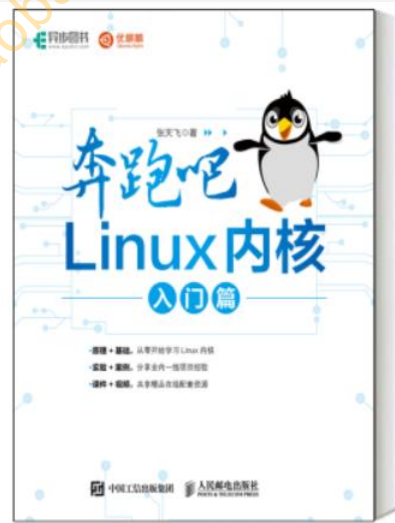


➤ 《奔跑吧Linux内核》入门篇适合人群

- ✓ 已经阅读了鸟哥私房菜，想进一步提升Linux系统的理解能力
- ✓ 奔跑卷没有满分的
- ✓ 喜欢Linux和开源软件的爱好者
- ✓ Linux以及计算机相关的从业人员，如销售，PM，FAE以及研发人员等。
- ✓ 喜欢计算机的在校大学生以及操作系统课程教辅教材
- ✓ 中小学生电脑极客



进一步提升自己



入门篇内容介绍

- 第一章 Linux系统入门
- 第二章 Linux内核基础知识
- 第三章 内核编译和调试
- 第四章 内核模块
- 第五章 简单的字符设备驱动
- 第六章 系统调用

入门篇内容介绍

- 第七章 内存管理
- 第八章 进程管理
- 第九章 同步管理
- 第十章 中断管理
- 第十一章 调试和性能优化
- 第十二章 开源社区

Linux系统简介

从Linux小白到Linux系统专家
只差一本《奔跑吧Linux内核》
<https://shop1156836.taobao.com>

Linux系统无处不在

- Linux基金会2017年发布的一组数据：
 - ✓ 90%的公有云应用在使用Linux系统
 - ✓ 62%的嵌入式市场在使用Linux系统
 - ✓ 99%的超级计算机在使用Linux系统
 - ✓ 82%的手机操作系统在使用Linux系统
 - ✓ 全球100万个顶级域名中超过90%都在使用Linux
 - ✓ 全球大部分的股票交易市场都是基于Linux系统来部署的
 - ✓ 全球知名的淘宝网、亚马逊网、易趣网、沃尔玛等电子商务平台都在使用Linux

从Linux/Android/iOS内核入门
只差一本《Linux/Android/iOS内核入门》
<https://shop115024135.taobao.com>

Linux发展历史

- Linux诞生于1991年10月5日，它的产生和开源运动有着密切的关系。
 - ✓ 1983年，Richard Stallman发起GNU（GUN's Not UNIX）计划
 - ✓ 1991年，Linus Torvalds在一台386电脑上学习Minix操作系统，并动手实现了一个新的操作系统，然后在comp.os.minix新闻组上发布了第一个版本的Linux内核。
 - ✓ 1993年，Linux 0.99的代码已经有大约10万行。
 - ✓ 1994年，采用GPL（General Public License）协议的Linux 1.0正式发布
 - ✓ 1995年，Bob Young创办了Red Hat公司，以GNU/Linux为核心，把当时大部分的开源软件打包成一个发行版，这就是RedHat Linux发行版。
 - ✓ 1996年，Linux 2.0发布，该版本可以支持多种处理器，如alpha、mips、powerpc等，内核代码量大约是40万行。

- Linux诞生于1991年10月5日，它的产生和开源运动有着密切的关系。
 - ✓ 1999年，Linux 2.2发布，支持ARM处理器。第一家国产Linux发行版——蓝点Linux系统诞生，它是第一个支持在帧缓冲上进行汉化的Linux中文版本。
 - ✓ 2001年，Linux 2.4发布，支持对称多处理器SMP和很多外设驱动。同年，毛德操老师出版了《Linux 2.4内核源代码情景分析》。
 - ✓ 2003年，Linux 2.6发布。与Linux 2.4相比，该版本增加了很多性能优化的新特性，使它成为真正意义上的现代操作系统。
 - ✓ 2008年，谷歌正式发布Android 1.0，Android系统基于Linux内核来构建。在之后的十年里，Android系统占据了手机系统的霸主地位。
 - ✓ 2011年，Linux 3.0发布。在长达8年的Linux 2.6开发期间，众多IT巨头持续为Linux内核贡献了很多新特性和新的外设驱动。同年，全球最大的Linux发行版厂商Red Hat宣布营收达到10亿美元。
 - ✓ 2015年，Linux 4.0发布。

Linux发行版 – Red Hat

➤ Fedora Core

- ✓ Fedora Core发行版是Red Hat公司的新技术测试平台，很多新的技术首先会应用到Fedora Core上，等测试稳定了才会加入Red Hat的RHEL版本中

➤ RHEL (Red Hat Enterprise Linux)

- ✓ RHEL是面向服务器应用的Linux发行版，注重性能、稳定性和服务器端软件的支持。
- ✓ 2018年4月Red Hat公司发布的RHEL 7.5操作系统，提升了性能，增强了安全性。

➤ CentOS (Community Enterprise Operating System)

- ✓ 根据RHEL的源代码重新编译而成
- ✓ CentOS不包含封闭源代码的软件
- ✓ CentOS可以免费使用，并由社区主导

Linux发行版 – Debian系列

➤ Debian Linux

- ✓ Debian由Ian Murdock在1993年创建，是一个致力于创建自由操作系统的合作组织。因为Debian项目以Linux内核为主，所以Debian一般指的是Debian GNU/Linux。Debian能风靡全球的主要原因是其特有的apt-get/dpkg软件包管理工具，该工具被誉为所有Linux软件包管理工具中最强大、最好用的一个。

➤ Ubuntu Linux

- ✓ Ubuntu的中文音译是“乌班图”，是以Debian为基础打造的以桌面应用为主的Linux发行版。Ubuntu注重提高桌面的可用性以及安装的易用性等方面，因此经过这几年的发展，Ubuntu已经成为最受欢迎的桌面Linux发行版之一。

Linux发行版 – SuSE

➤ SuSE Linux

- ✓ SuSE Linux是来自德国的著名Linux发行版，在Linux业界享有很高的声誉。SuSE公司在Linux内核社区的贡献仅次于Red Hat公司，培养了一大批Linux内核方面的专家。SuSE Linux在欧洲Linux市场中占有将近80%的份额，但是在中国的市场份额并不大。

从Linux小白到Linux大神
只差一本《奔跑吧Linux入门篇》
<https://shop115683645.taobao.com/>

Linux发行版 – 国产Linux发行版

➤ 优麒麟Linux

- ✓ 优麒麟 (Ubuntu Kylin) Linux诞生于2013年，是由中国国防科技大学联合Ubuntu、CSIP开发的开源桌面Linux发行版，是Ubuntu的官方衍生版。该项目以国际社区合作方式进行开发，并遵守GPL协议。

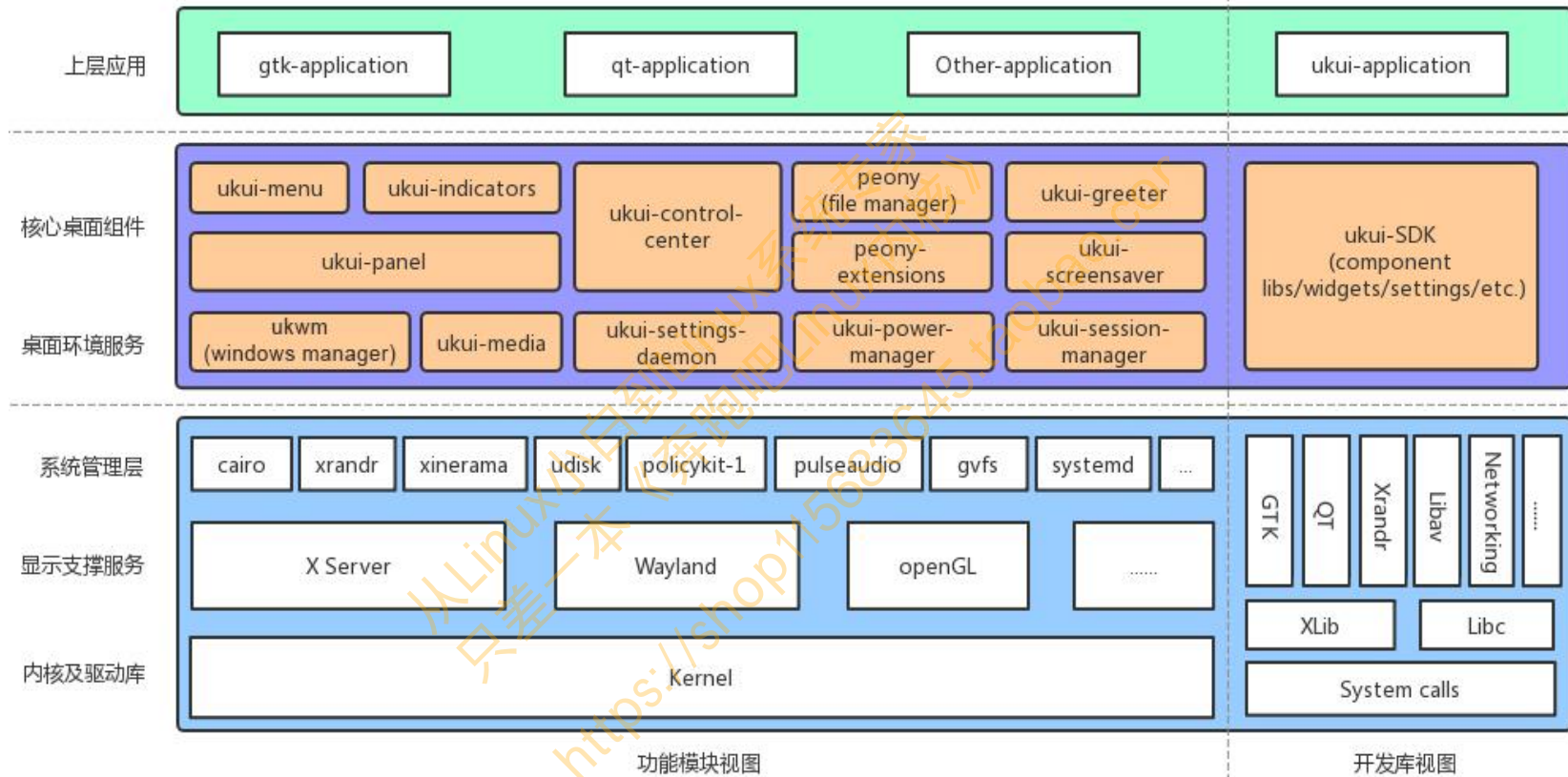


Linux发行版 – 国产Linux发行版

➤ 优麒麟Linux

- ✓ 优麒麟自研的UKUI轻量级桌面环境，按照Windows用户的使用习惯进行设计开发，开创性地将Windows标志性的“开始”菜单、任务栏引入Linux操作系统中，并且对文件管理器、控制面板等桌面重要组件进行了深度定制，减少了Windows用户迁移到Linux平台的时间成本，具有稳定、高效、易用的特点。
- ✓ 优麒麟Linux默认安装的麒麟软件中心、麒麟助手、麒麟影音、WPS办公软件、搜狗输入法等软件让普通用户更易上手。针对ARM平台的安卓原生兼容技术，优麒麟可以把安卓强大的生态软件无缝移植到Linux系统中。基于优麒麟Linux的银河麒麟企业发行版，支持x86和ARM64架构，在中国的政企市场有较大的占有率。

UKUI架构设计图



微内核与宏内核

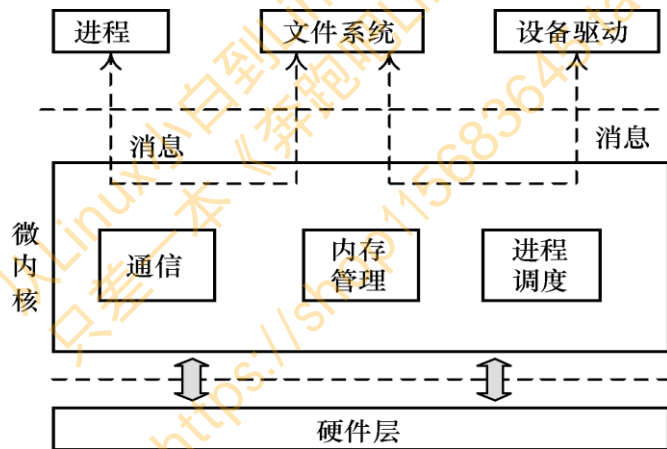
从Linux小白到Linux系统专家
只差一本《奔跑吧Linux内核》
<https://shop115683645.taobao.com>

操作系统的主要功能

- 操作系统属于软件的范畴，来负责管理系统的硬件资源，同时为应用程序开发和执行提供配套环境。
 - ✓ 为多用户和应用程序管理计算机上的硬件资源。
 - ✓ 为应用程序提供执行环境。
 - ✓ 并发性：操作系统必须具备执行多个线程的能力。从宏观上看，多线程会并发执行，如在单CPU系统中运行多线程的程序。线程是独立运行和独立调度的基本单位。
 - ✓ 虚拟性：多进程的设计理念就是让每个进程都感觉有一个专门的处理器为它服务，这就是虚拟处理器技术。

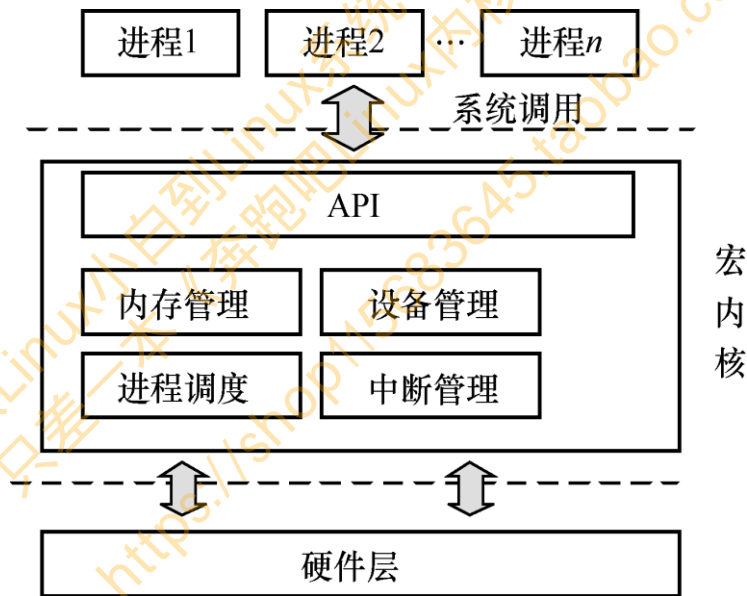
微内核

- 操作系统内核的设计在历史上存在两大阵营，一个是宏内核，另一个是微内核
- 微内核：把操作系统分成多个独立的功能模块，每个功能模块之间的访问需要通过消息来完成



宏内核

- 宏内核：指所有的内核代码都编译成一个二进制文件，所有的内核代码都运行在一个大内核地址空间里，内核代码可以直接访问和调用，效率高并且性能好。



Linux设计之道

- Linux在设计之初并没有使用当时学术界流行的微内核架构，而是采用实现方式比较简单的宏内核架构。
 - ✓ Linux在当时是业余作品
 - ✓ Linus本人更喜欢宏内核的设计
 - ✓ 宏内核架构的优点是设计简洁和性能比较好
 - ✓ 微内核架构最大的问题就是高度模块化带来的交互的冗余和效率的损耗
- 所有的理论设计放到现实的工程实践中都是一种折中的艺术。
- Linux内核融合了宏内核和微内核的优点。
 - ✓ 模块化设计
 - ✓ 抢占式内核、
 - ✓ 动态加载内核模块

Linux内核开发模式

从Linux小白到Linux系统专家
只差一本《奔跑吧Linux内核》
<https://shop115683645.taobao.com>

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

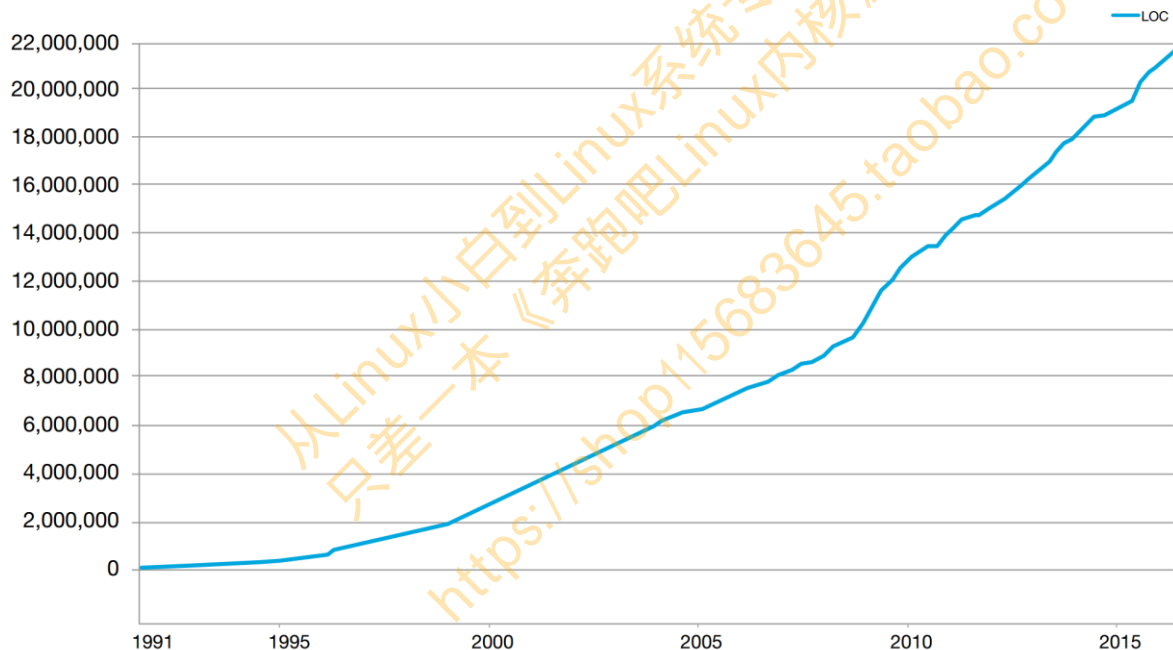
Kernel Version	Release Date	Days of Development
4.4	2016-01-10	70
4.5	2016-03-13	63
4.6	2016-05-15	63
4.7	2016-07-24	70

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%
Bartłomiej 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:

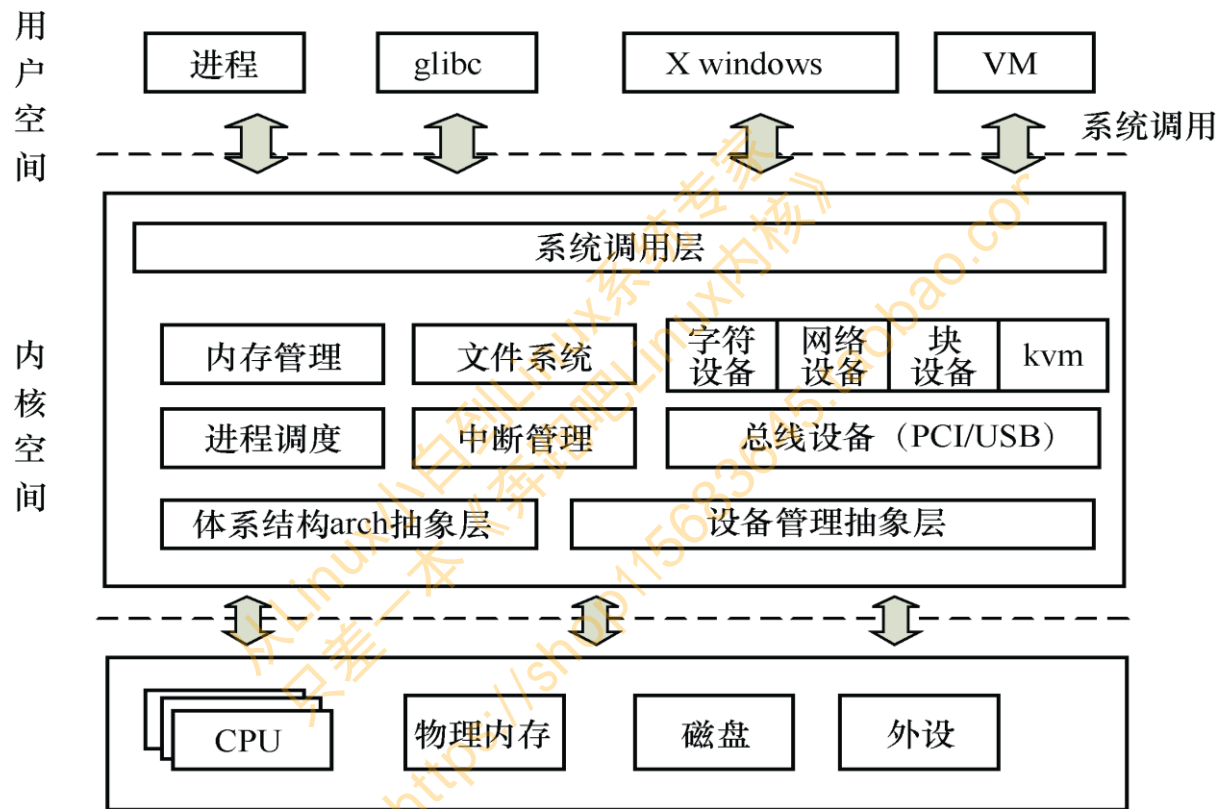
Company	Changes	Percent
Intel	14,384	12.9%
Red Hat	8,987	8.0%
none	8,571	7.7%
unknown	7,582	6.8%
Linaro	4,515	4.0%
Samsung	4,338	3.9%
SUSE	3,619	3.2%
IBM	2,995	2.7%
consultants	2,938	2.6%
Renesas Electronics	2,239	2.0%
Google	2,203	2.0%
AMD	2,100	1.9%
Texas Instruments	1,917	1.7%
ARM	1,617	1.4%
Oracle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

Linux内核OverView

从Linux小白到Linux系统专家
只差一本《奔跑吧Linux内核》
<https://shop115683645.taobao.com>

Linux内核架构概貌



Linux内核架构OverView

➤ 系统调用层

- ✓ Linux内核把系统分成两个空间：用户空间和内核空间
- ✓ Linux内核为内核态和用户态之间的切换设置了软件抽象层，叫作系统调用（System Call）层，其实每个处理器体系结构设计中都提供了一些特殊的指令来实现内核态和用户态之间的切换。Linux内核充分利用了这种硬件提供的机制来实现系统调用层。
- ✓ 系统调用层最大的目的是让用户进程看不到真实的硬件信息。
- ✓ 当用户需要读取一个文件的内容时，编写用户进程的程序员不需要知道这个文件具体存放在磁盘的哪个扇区里，只需要调用open()、read()或mmap()等函数即可。

Linux内核架构OverView

➤ 处理器体系结构抽象层

- ✓ Linux最初的设计只支持x86体系结构，后来不断扩展，到现在已经支持几十种体系结构
- ✓ Linux内核为不同体系结构的实现做了很好的抽象和隔离，也提供了统一的接口来实现。比如，在内存管理方面，Linux内核把和体系结构相关部分的代码都存放在arch/xx/mm目录里，把和体系结构不相关的代码都存放在mm目录里，从而实现完好的分层。

Linux内核架构OverView

➤ 进程管理

- ✓ 进程是现代操作系统中非常重要的概念，包括上下文切换（Context Switch）以及进程调度（Scheduling）
- ✓ 每个进程运行时都感觉完全占有了全部的硬件资源，但是进程不会长时间占有硬件资源
- ✓ 操作系统利用进程调度器让多个进程并发执行
- ✓ Linux内核并没有严格区分进程和线程，而常用task_struct数据结构来描述
- ✓ Linux内核的调度器的发展经历了好几代，从很早的O(n)调度器到Linux 2.6内核中的O(1)调度器，再到现在的CFS公平算法调度器
- ✓ 目前比较热门的讨论是关于性能和功耗的优化，比如ARM阵营提出了大小核体系结构，至今在Linux内核实现中还没有体现，因此类似EAS（Energy Awareness Scheduling）这样的调度算法是一个研究热点

Linux内核架构OverView

➤ 内存管理

- ✓ 内存管理模块是Linux内核中最复杂的模块，它涉及物理内存的管理和虚拟内存的管理
- ✓ 虚拟内存的模块有反向映射、页面回收、KSM、mmap映射、缺页中断、共享内存、进程虚拟地址空间管理等
- ✓ 物理内存管理包括：物理内存初始化，页面分配器（Page Allocator），伙伴系统，slab分配器等等。

从Linux小白到内核专家
只差一本《奔跑吧Linux内核》
<https://shop115683645.taobao.com>

Linux内核架构OverView

➤ 中断管理

- ✓ 中断管理包含处理器的异常 (Exception) 处理和中断 (Interrupt) 处理
- ✓ 异常通常是指如果处理器在执行指令时检测到一个反常条件，处理器就必须暂停下来处理这些特殊的情况，如常见的缺页异常 (Page Fault)
- ✓ 中断异常一般是指外设通过中断信号线路来请求处理器，处理器会暂停当前正在做的事情来处理外设的请求
- ✓ Linux内核在中断管理方面有上半部和下半部之分。上半部是在关闭中断的情况下执行的，因此处理时间要求短、平、快；而下半部是在开启中断的情况下执行的，很多对执行时间要求不高的操作可以放到下半部来执行。
- ✓ Linux内核为下半部提供了多种机制，如软中断、Tasklet和工作队列等。

Linux内核架构OverView

➤ 设备管理

- ✓ 设备管理对于任何一个操作系统来说都是重中之重。Linux内核之所以这么流行，就是因为它支持的外设是所有开源操作系统中最多的。当很多大公司有新的芯片诞生时，第一个要支持的操作系统是Linux，也就是尽可能地在Linux内核社区里推送。
- ✓ Linux内核的设备管理是一个很广泛的概念，包含的内容很多，如ACPI、设备树、设备模型kobject、设备总线（如PCI总线）、字符设备驱动、块设备驱动、网络设备驱动等。

从Linux小白到Linux大神
只差一本《奔跑吧Linux》
<https://shop115683615.taobao.com>

Linux内核架构OverView

➤ 文件系统

- ✓ 为了支持各种各样的文件系统，Linux抽象出了一个称为虚拟文件系统（VFS）层的软件层，这样Linux内核就可以很方便地集成多种文件系统。
- ✓ 一个优秀的操作系统必须包含优秀的文件系统，但是文件系统有不同的应用场合，如基于闪存的文件系统F2FS、基于磁盘存储的文件系统ext4和XFS等

从Linux小白到Linux内核入门
只差一本《奔跑吧Linux内核》
<https://shop115683645.taobao.com/>

Linux内核之学习方法

➤ C语言是基本功

- ✓ 熟悉C语言是学习Linux内核的基础。读者可以重温C语言课程，然后阅读一些经典的C语言著作，如《C专家编程》《C陷阱和缺陷》和《C与指针》等。

➤ 安装和使用Linux发行版，如优麒麟Linux 18.04。

- ✓ 熟悉常用的Linux开发工具，比如Vim和Git等。

➤ 在Linux机器上做一些编程和调试的练习

- ✓ 使用笨叔提供“00”编译内核的git tree，结合QEMU+GDB+Eclipse来单步调试内核
- ✓ 练习本书的实验
- ✓ 编写简单的设备驱动

➤ 研究Linux内核的一些核心API的实现

➤ 从项目中深入学习

➤ 遇到问题不要轻易放弃，以问题为导向，刨根问底。可以结合蓝色版本的《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》

<https://shop115683645.taobao.com>

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

奔跑吧
LINUX社区

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

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

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

奔跑2.14	匿名页面的生命周期	0:26:16
--------	-----------	---------

2.15 页面迁移

奔跑2.15	页面迁移	0:19:07
--------	------	---------

2.16 内存规整

奔跑2.16	内存规整	0:24:03
--------	------	---------

2.17 KSM

奔跑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
上集x86_64	
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
下集arm64	
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: Runing-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

<https://shop11538560.com>