

LinuxLinux 内核Troubleshooting

关注者521

被浏览98,612

你为什么学习 Linux 内核?

如题，你学习 Linux 内核的原因是什么，想实现什么样的效果，有什么好的学习方法

关注问题

写回答

邀请回答


添加评论

分享

举报

...

查看全部 27 个回答

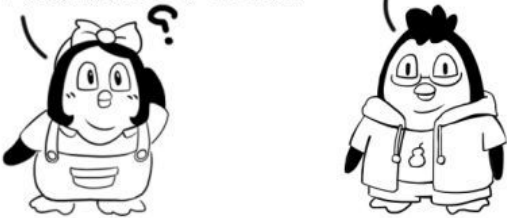


**嵌入式Linux**  
公众号「嵌入式Linux」,嵌入式入门和行业咨询

264 人赞同了该回答

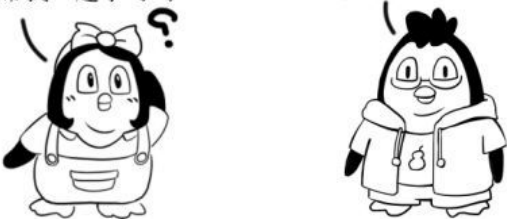
前言

嗯，那个大嘴巴传得这么快  
篮球，听说你最近在学习Linux?



你能带我一起学习吗?

你每天给我带个早餐  
我可以考虑一下，以后毕业了工资上万哦



Accounting English

復旦大學出版社

托福

关于作者



**嵌入式Linux**  
公众号「嵌入式Linux」,...

回答905

文章189

关注者12,915

关注他

发私信

被收藏 575 次

- 学习啦

7 人关注

林小羊 创建
- 编程

4 人关注

代码魔女lital 创建
- 奇怪

2 人关注

代码魔女lital 创建
- 看起来非常厉害

1 人关注

Carl Xu 创建
- 学习

1 人关注

liushiqi9 创建

相关问题

- 为什么Linux内核代码质量管理，容忍内核编译过程中使用默认编译选项时，编译器(gcc)警告的海量Warning?

6 个回答
- 为什么linux内核崩溃会导致重启，而应用程序崩溃不会导致重启?

7 个回答
- Linux 内核4.0 有什么亮点?

5 个回答
- 为什么linux内核要压缩?

6 个回答



1.1 随便先说下

Linux 内核非常庞大，我说的非常大并不是为了吓唬大家，确实是非常多的代码，超过 600 万行的代码，所以我写文章介绍 Linux 内核，也不可能每一行代码去分析，但是我会提炼其中的重点出来，告诉大家，Linux 内核的构成，包含哪些东西，我们不管学习什么，最关键的是学会其中的思想，但是我们如果什么都还不会的时候，可以学着由表透里，就像我打篮球一直不会后仰跳投，但是我可以把科比的研究一遍又一遍，总有一天我也能听到自己打铁的哐哐声。

但是我希望我的文字略带微笑着去面对这些代码，去面对 Linux 森林，然后你从这个森林走出来后，可以明白森林里的有哪些路，你下一次想带个妹子进去约会，可以找到属于自己的岔路。

虽然 Linux 内核非常庞大，但是横向对比计算机的发展史，Linux 的历史非常的简短，计算机出现的时候，还没有什么花里胡哨的包装，都是光着身子跑，用技术的语言说就是它只能执行一个任务，执行一个进程，也可以叫“裸跑”，学习计算机的同学应该很快明白什么是裸机程序，裸机程序只可以让一个进程使用硬件资源，这无形上对硬件资源的浪费。

可以拿我们的手机来比喻，裸机的话，我们玩王者荣耀的时候，有人打电话进来的话，王者荣耀是要被终止的，后来就有了操作系统，操作系统可以保证多任务执行，同样的 CPU 芯片，有了操作系统，我可以开微信，QQ，还同时做其他很多事情。

所以就出来了很多操作系统，操作系统我认为可以分为实时操作系统和通用操作系统，实时操作（RTOS 全称 Real Time Operating System）系统可以理解为对时间要求非常苛刻，可以用一个词“必须”来理解，就是在某个时间片之内必须要做某件事情。

而通用操作系统，我们可以很常见，比如电脑 windows，Android 手机等，Linux 也可以认识是通用 OS，因为他们对时间上要求不是很严苛。



Linux 可以通过配置某个宏定义变成实时操作系统，但是我们使用 Linux 主要是针对他的通用 OS，多用户，多任务等特别突出的性能。

世界上的嵌入式操作系统数不胜数，我有一个很厉害的师弟，前几年也参与了一个国内嵌入式操作系统的开发，当然了，他们是以盈利为目的了，还有我认识的周立功先生，他们公司也研发了一个嵌入式操作系统，但是对于我们开发者来说，做项目的时候，选择适合自己项目的系统就好了，对于学习来说，我认为，应该学习最先进的操作系统，这样才能让自己的技术有更大的先进性。

### VxWorks

VxWorks 是美国 WindRiver 公司的产品，市场占有率比较高的嵌入式操作系统。VxWorks 实时操作系统由 400 多个相对独立、短小精悍的目标模块组成，用户可根据需要选择适当的模块来裁剪和配置系统，具有优先级的任务调度、任务间同步与通信、中断处理、定时器和内存管理等功能，符合 POSIX (可移植操作系统接口)规范的内存管理，多处理器控制程序，并且具有简明易懂的用户接口，在核心方面甚至可以微缩到 8 KB。

这个操作系统不敢吹太多，可以自己去了解下，因为稳定性太好了，应用到了航空，卫星，军事等各个领域，实用性非常强，所以很多培训机构也针对这个操作系统开了培训课程。

### $\mu$ C / OS-II

$\mu$ C / OS-II是在 $\mu$ C-OS的基础上发展起来的，是美国嵌入式系统专家 Jean J. Labrosse 用 C 语言编写的一个结构小巧、支持抢占式的多任务实时内核。 $\mu$ C / OS-II 能管理 64 个任务，并提供任务调度与管理、内存管理、任务间同步与通信、时间管理和中断服务等功能，具有执行效率高、占用空间小、实时性能优良和可扩展性强等特点。

### $\mu$ Clinux

$\mu$ Clinux 是一种优秀的嵌入式 Linux 版本，其全称为 micro-control Linux，从字面意思看是指微控制 Linux。同标准的 Linux 相比， $\mu$ Clinux 的内核非常小，但是它仍然继承了 Linux 操作系统的主要特性，包括良好的稳定性和移植性、强大的网络功能、出色的文件系统支持、标准丰富的 API，以及 TCP / IP 网络协议等。因为没有 MMU 内存管理单元，所以其多任务的实现需要一定技巧。

### eCos

eCos(embedded Configurable operating system)，即嵌入式可配置操作系统。它是一个源代码开放的可配置、可移植、面向深度嵌入式应用的实时操作系统。最大特点是配置灵活，采用模块化设计，核心部分由小同的组件构成，包括内核、C 语言库和底层运行包等。每个组件可提供大量的配置选项(实时内核也可作为可选配置)，使用 eCos 提供的配置工具可以很方便地配置，并通过不同的配置使得 eCos 能够满足不同的嵌入式应用要求。

你学了Linux后，就会觉得其他系统也很容易上手  
这么多系统，学来学去，我都老了





内核的工作

我们使用的计算机大家都知道是操作系统，那内核是什么呢？那我们先简单说说操作系统，操作系统是面向用户的，计算机用户可以使用计算机操作系统来工作，聊天，玩游戏，我们使用的这些东西都是应用软件，对应用程序来说，内核就是它的操作系统，这个系统可以为应用程序工作，管理应用程序。

内核还有一个比较重要的工作，就是管理应用，为应用程序准备好运行内存，管理应用程序的执行，让应用程序通行无阻，当然了，这个是理想的情况，不理想的情况就是导致内存或者资源不够用的情况，应用出现崩溃等异常。

除了管理应用以外，内核还需要管理硬件设备，Linux 内核下面有非常多的设备驱动代码，如果一个内核开发工程师说他不懂设备驱动，那简直就是一个笑话，内核跟 CPU 和硬件设备关系非常密切，在整个操作系统中的地位，具有承上启下的作用。

1.2 UNIX 的诞生

生日：UNIX 在 1969 年出生。

他的父亲和母亲：是 Dennis Ritchie 和 Ken Thompson 两个人擦出了灵感的火花创造出来的。

出生户籍地址：贝尔实验室



贝尔实验室图片



贝尔实验室的logo

#### 出生具体流程：

1965 年，贝尔实验室要做一个项目，这个项目叫PDP-7计算机计划，发起人是通用电气和麻省理工学院，他们给这个操作系统起了一个漂亮的名字叫做「**MULTICS 操作系统**」（"Multiplexed Information and Computing Service"的缩写）。做事情总是有个计划，他们给这个操作系统给出的计划是，这个操作系统可以多个人使用，按照我们现在的人来说就是多用户系统，多任务，多层次等等。

到了1969 年，发起人觉得这个进度太慢了，本来想早点制造出来我们好用来玩电脑游戏的，结果你们这几个科学家整了这么久还是没整出来，那只好停掉了，停掉了投资方就不再提供后备的资源了，留下的东西就自己瞎整吧，投资方也不管了。

计划被停下来了，当时，Ken Thompson 在调试一个程序，这个程序名字叫做“星级旅游”，这个程序运行在一个叫做 GE-635 的机器上面，但是因为这个机器的硬件设备比较落后，运行速度非常慢，这让Ken Thompson感觉非常不爽，然后他发现之前做「PDP-7计算机计划」项目的时候还有一台PDP-7计算机，这个计算机就是图片下面的那个计算机，当时应该没有人想到计算机可以做到这么小，然后他们就把 GE-635 程序移植到 PDP-7 计算机上面。

到了1970年，PDP-7 可以运行 GE-635程序了，但是却只能支持两个用户，当时 Brian Kernighan 就开玩笑的称他们的系统是“UNIpIexed Information and Computing Service”，这个缩写就是 UNICS，再后来，大家就取谐音，称为 UNIX。所以1970 年可以称为 UNIX元年。

汤姆逊和丹尼斯里奇







PDP-7计算机

### 1.3 BSD操作系统

伯克利软件套件（英语：Berkeley Software Distribution，缩写为 BSD），也被称为伯克利 UNIX（Berkeley UNIX），是一个操作系统的名称。衍生于UNIX（类UNIX），1970年代由伯克利加州大学的学生比尔·乔伊（Bill Joy）开创，也被用来代表其衍生出的各种套件。

BSD 常被当作工作站级别的 UNIX 系统，这得归功于 BSD 用户许可证非常地宽松，许多 1980 年代成立的计算机公司，不少都从 BSD 中获益，比较著名的例子如 DEC 的 Ultrix，以及 Sun 公司的 SunOS。1990 年代，BSD 很大程度上被 System V 4.x 版以及 OSF/1 系统所取代，但其开源版本被采用，促进了因特网的开发。

BSD 比 Linux 早出现，稳定性和安全性都在 Linux 之上，甚至 Windows 和 OS X 都有来自 BSD 的代码，但是现在一提到开源自由软件，人们首先想到的是Linux，而不是资格更老的BSD。

UNIX创始人之一的 Ken Thompson 曾如此评价 Linux，「Linux不过是反微软思潮下的产物」，这个家伙觉得 Linux 不可能有多大的成就，非常自信的觉得 BSD 在任何时候都可以击败 Linux，甚至觉得Linux 是一个低端和下流的操作系统内核。

但是事实证明，Linux 赢得了这场战争，有实力，也有些运气，Linux 在发展的时候，BSD 当时正被官司缠上，没有多余的心思应战 Linux，而BSD腹背受敌的原因很明显是因为他们的商业化，而Linux正是因为开源而更受全世界开发者的青睐。

一个事情的成功，90% 是由他的领导者决定的，就好像一个球队能走多远，队长和教练可以决定它的深度，Linux 也一样，Linus Torvalds 是位杰出的领袖人物，他成功的让一群性格迥异的、绝非泛泛之辈的黑客共同合作开发，而没有如其他开源项目一般分崩离弃。

还有一点，Linux 的硬件支持比 BSD 好，这在各种终端设备上来说简直就是一种惊喜，随着PC不断被边缘化，移动端的产品急需找到一款适合他们的操作系统，Linux 是最好不过的人选了。

**GNU 的大力支持**，GNU 的许可证与 BSD 不兼容，因此 Linux 的出现让两者完美结合，所以现在 Linux 全名叫 GNU/Linux。



BSD 走的是教堂式的学院派路线，而 Linux 则是代表了市集式的骇客精神，BSD 很自信，觉得自己是贵族出生，应该享受这世界上的荣华富贵，而 Linux 再一次验证了农村包围城市的必然性，成王败寇，功败垂成。



BSD 原本就有极佳的根基，缺乏的可能是一点运气，未来或许大有可为。

(点击查看大图)

#### 1.4 GNU计划的产生





理查·斯托曼

因为 UNIX 操作系统的商业化，原来的 UNIX 系统已经不能再被随意的使用，很多人都希望能有一款免费好用的操作系统，因为并不是每个人都很有钱，也不是每个人都有能力自己去写操作系统，此时，理查·斯托曼在麻省理工学院人工智能实验室发起 GNU 计划，希望发展出一套完整的开放源代码操作系统来取代 UNIX，计划中的操作系统，名为 GNU。

1983年9月27日，理查·斯托曼在 net.UNIX-wizards 和 net.usoft 新闻组中公布这项计划。在此项计划中，开发出了很多我们现在熟悉的常用的工具，包括GNU编译器套装（GCC）、GNU的C库（glibc）、以及 GNU 核心工具组（coreutils）。另外也是 GNU 除错器（GDB）、GNU 二进制实用程序（binutils）的 GNU Cash shell中和 GNOME 桌面环境。

当然，GNU计划的目的还是开发出一款自由传播的操作系统，这个操作系统的名字叫 Hurd，但是由于对操作系统的要求过高，以至于 Hurd 一直处于测试阶段，本意是一个好事情，但是能力有限啊，开发的东西老是出bug，再好的创意那也是徒劳了。

不过 Linus 大神通过 GNU 发布了自己的 Linux 系统后，就火起来了，真的就一发不可收拾，这也是为什么 GNU 和 Linux 关系密切的原因。

### 1.5 UNIX 衍生系统发展图

用文字来描述事实总是感觉有点欠缺，就好像两个人发生争执，可以通过吵嘴解决问题，也可以通过大家解决问题，但是我认为打架应该是最直接的，你说得再多也没有枪杆子来得实在。

本书的重心主要放在 Linux 上，可以观察 Linux 的发展轨迹，还是非常给力的，当然了，BSD 目前来说市场占有率不能跟 Linux 相提并论，但是他在整个 UNIX 上也有着有自己的一席之地的。







## 1.6 Linux的导火索MINIX

### MINIX启动界面

在 UNIX 产生后，版权在 AT&T 手里，在 Version 7 UNIX 发布之后，发布新的授权条款，将 UNIX 源码私有化，在大学不得再使用 UNIX 源码，荷兰阿姆斯特丹自由大学计算机科学系的塔能鲍姆教授（Andrew Stuart "Andy" Tanenbaum）为了教学,自己写了一个类 UNIX 的小系统，命名为 MINIX（意为mini-UNIX）。





永远不要小看任何一个人，如果这个人能够编写出自己的教学操作系统，你更加不要随便惹他，你可能不可以，但是下面的这个家伙是可以的，大家可能都不知道什么是“宏内核”和“微内核”，但是这个家伙和 Linus 的辩论轰动一时，不管怎么说，Linux 应该是现在的胜利者，最直接的原因是开源，让更多的开发者可以使用 Linux 内核移植到自己的设备上，包括 ARM 设备。

但是我们不能抹杀 ast 的作用，在计算机系统的贡献上，和教学的贡献上，肯定是具有一席之地的，作为本文的撰写者，他们都是我们的始祖，技术无国界，请收下我的膝盖。

## 1.7 Linux 的出生

### 林纳斯 托瓦兹 (Linus Torvalds)

我们的主角人物，林纳斯 托瓦兹 (Linus Torvalds) 1991年，林纳斯·托瓦兹在赫尔辛基大学上学时，对操作系统很好奇。由于但是 386BSD 还没有出来。可是他不喜欢他的 386 电脑上的 MS-DOS 操作系统，所以就安装了 Minix，可对 Minix 只允许在教育上使用很不满（在当时 Minix 不允许被用作任何商业使用），于是他便开始写他自己的操作系统。

Linux 的第一个版本在 1991 年 9 月被大学 FTP server 管理员 Ari Lemmke 发布在 Internet 上，最初 Torvalds 称这个内核的名称为「Freax」，意思是自由「free」和奇异「freak」的结合字，并且附上「X」这个常用的字母，以配合所谓的类 UNIX 的系统。但是 FTP 服务器管理员嫌原来的命名「Freax」的名称不好听，把内核的称呼改成「Linux」，当时仅有 10000 行代码，仍必须运行于 Minix 操作系统之上，而且必须使用硬盘开机，随后在 10 月份第二个版本 (0.02 版) 发布，同时这位芬兰赫尔辛基的大学生在 comp.os.minix 上发布这样一则公告



Hello everybody out there using minix- I'm doing a (free) operation system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones.



1994 年 3 月, Linux1.0 版正式发布。为了让 Linux 可以在商业上使用, 林纳斯·托瓦兹决定更改他原来的协议 (这个协议会限制商业使用), 以 GNU GPL 协议来代替。之后许多开发者致力融合 GNU 元素到 Linux 中, 做出一个有完整功能的、自由的操作系统。

80386 的芯片

80386 的电脑

如果单凭林纳斯一个人的力量, Linux 不可能发展到这个程度, 我认为在那个时候, 他做了一个非常正确的决定, 就是「开源」, 让世界上更多的优秀程序员加入到他的事业当中, 为了让更多的人同步开发, 林纳斯还写了 GIT, 这个让很多协作开发者为之兴奋的工具。

### 1.8 Linux 的标志物





Linux 的标志和吉祥物是一只名字叫做 Tux 的企鹅，标志的由来是因为 Linus 在澳洲时曾被一只动物园里的企鹅咬了一口，便选择企鹅作为 Linux 的标志。更容易被接受的说法是：企鹅代表南极，而南极又是全世界所共有的一块陆地。这也就代表 Linux 是所有人的 Linux。

曾经有一个笑话说明纳斯被企鹅咬了之后，因为咬过的伤口会发炎，发炎的时候伤口会有点疼，晚上写代码想打瞌睡，但是就是因为这个炎症的疼痛感刺激着自己，当然了，这个只是个传说，传说是否是真的，哪天大神心情好了可能会揭晓答案。

### 1.9 Linux的现状

今天在 Linus Torvalds 带领下，众多开发共同参与开发和维护 Linux 内核。理查德·斯托曼领导的自由软件基金会，继续提供大量支持 Linux 内核的 GNU 组件。一些个人和企业开发的第三方的非 GNU 组件也提供对 Linux 内核的支持，这些第三方组件包括大量的作品，有内核模块和用户应用程序和库等内容。Linux 社区或企业都推出一些重要的 Linux 发行版，包括 Linux 内核、GNU 组件、非 GNU 组件，以及其他形式的软件包管理系统软件。

目前这个阶段，可以说每个人都脱离不开 Linux，好吧，肯定有人跑出来抬杠，我就问你，你手机底层是 Linux 内核你可知道，你说你用的是塞班手机，那里购物的云平台，淘宝，亚马逊等都是用 Linux 开发维护的，好吧你说你是个老板，买东西都是别人给你买的，那么你炒股吧，很多股票平台都是基于 Linux 开发维护的。





### 1.10 为什么学习Linux

Linux 内核现在覆盖的领域非常广，手机、平板、路由器等等，就大家非常喜欢的苹果操作系统，底层内核也是有 Linux的影子，Linux 的普及毋庸置疑，学习 Linux 应该作为每个技术人员的标配。

刚开始参加工作的时候，很多面试官都问我，你对 Linux 是不是非常懂，我每次都回答，我对 Linux 也是刚刚入门，但是我非常喜欢Linux，而且我也会持续在 Linux 上面做研究，学习，我也喜欢和这些 Linux 爱好者一起探讨问题。

学习完 Linux 内核你会对整个计算机体系有一个更深刻的认知，作为一个开发者，不管你从事的是驱动开发，应用开发，还是后台开发，你都需要理解计算机操作系统和内核的运行机制，才可能更好的编写你的代码，出现更少的错误。

作为开发人员，不应该只局限在自己的小领域，因为你设计的模块，看起来非常小，但是你不了解进程的调用机制，不知道进程会阻塞，就绪，执行几个状态，你怎么可能编写好一个低容错率的代码呢？

### 后续

第一篇文章，后面还写了10来篇，第二章我会给我建立好的虚拟机，还有github代码仓库链接，想学习的同学可以一起加入进来，Linux 确实是非常不错，不管是不是做开发的，自己做运维或者其他算法验证部分，用到Linux 也是非常方便的。

这篇文章花了非常久的时间，技术专栏文章，我希望一周发一篇长文就好了，其他时间允许给我说点其他事情。





共勉~

=====

[weixin.qq.com/r/Y0N9ZaX...](https://weixin.qq.com/r/Y0N9ZaX...) (二维码自动识别)

=====

连载文章~希望大家多多支持~

编辑于 2019-07-24

▲ 赞同 264 ▼ ● 6 条评论 ↗ 分享 ★ 收藏 ♥ 喜欢 ... 收起 ^

### 更多回答



LeGrandK

困

61 人赞同了该回答

学linux 内核是因为工作需要啊. 客户要求修的bug, 要修好才行啊.  
比如最近两天突然我客户让我修个drm/i915的bug, 我就得乖乖的去看代码去.

综上所述, 为了钱.

好的学习方法啊? 我来说说, 首先, 跳过boot loader跳过所有汇编跳过所有体系结构相关代码. 那部分投入是几乎没有回报的. 会打击你学下去的兴趣和自信心.

其次, 什么0.12内核啊, 不能提高你的水平. 相信我. 直接去看毛德操. 在看毛德操之前, 我觉得内核代码那么多, 无从下手. 看完了毛德操(的一部分), 我才发现原来读代码只不过是耗费时间的工作, 有时间就可以读, 并不是什么高不可攀的.

然后, 不要尝试给上游发送什么格式修复, 注释的标点符号修复这类patch. 帮人家修复个标点符号绝对不是你学kernel的初衷. 不要在这方面浪费时间甚至沾沾自喜. 这种格式/标点符号修复的机会留给我就好了~

▲ 赞同 61 ▼ ● 22 条评论 ↗ 分享 ★ 收藏 ♥ 喜欢 ...



VizXu

程序猿, 攻城狮, 挨踢人士

41 人赞同了该回答

好久没看到新答案了, 实在忍不住回答一记.

我学习Linux内核一个原因是目前在工作上需要掌握内核相关的知识. 系统移植升级, 性能优化, 软硬件裁剪, 客制化, 安全加强, 基础软件trouble shooting, 都需要对Linux尤其是内核有一定的掌握. **但最重要的还是掌握内核能让我看问题看的更深刻.**

举个例子吧, 有次移植系统到一台新硬件上发现usb2.0的口无法识别usb3.0设备 (一般来说3.0的设备插在2.0的口上会默认加载2.0的驱动, 也就是说系统把这个设备当2.0的设备用, 只是io速度慢一点). 本来呢, 机器上有两个3.0的口和两个2.0的口, 我就想吧, 也不是大问题, 顶多你把键盘鼠标插在2.0的口上, 留下两个3.0的口这样就好啦. 满心欢喜发了release note, 结果客户死活不同意, 说bug就是bug, 一定得修.

我的内心是崩溃的, 因为知道像这种情况没个两三天trouble shooting那是不可能修的好的. 但是没办法只能硬着头皮看内核代码, 打log, insmod, rmmod各种动作搞一遍才搞明白怎么回事. 不过一番折腾下来也对usb模块理解的更深刻了, 知道xchi和echi的区别, 对usb core和host也认识更深了.

▲ 赞同 41 ▼ ● 11 条评论 ↗ 分享 ★ 收藏 ♥ 喜欢 ...





查看全部 27 个回答

