

[学习](#) [程序员](#) [Linux](#) [Linux 系统管理](#) [系统管理](#)关注者
3174被浏览
225803

如何更深入地学习 Linux?

我是一个在校的大二学生，对 Linux 非常感兴趣，以后想往 Linux 方向发展，现在能熟练的操作 Linux 系统，基本上熟悉了各种命令，也能写一些 shell 脚本，平时写代码都用 Linux，基本上可以独立搭建各种开发环境，C 语言、操作系统原理学的不错，但是学校并没有这样的课程安排，现在也很迷茫，不知道如何才能更加深入的学习 Linux，比如 Linux 内核，驱动开发或者运维等等，希望各位大神能给一些意见

[5 条评论](#) [分享](#) [邀请回答](#) [举报](#) [...](#)[收起](#) ^[关注问题](#)[写回答](#)

38 个回答

默认排序

**Tianheng Chen**

半吊子码农，操作系统

收录于 编辑推荐 · 657 人赞同了该回答

鉴于越来越多的人看到了答案，我觉得现在有必要更正一些用词错误和概念错误，非常感谢杨周的建议和斧正。

增加内容：关于楼主背景，很多人问我关于我的背景问题，在此统一作答，楼主目前大学本科三年级，在美国上学，专业为ECE和数学，所以找实习的机会比较多，找到的概率也比国内同胞们大一些，我高中同班同学有两个人大二的时候就找到google实习的。本人学习成绩略渣，这个不是谦虚，是正儿八经的渣，GPA全系中位数，另：谁给我发私信，说大神膜拜什么的，会让我浑身不舒服，我认为高斯欧拉傅里叶可以称作大神，普通人嘛，就安安分分做普通人吧。

很多大大二的朋友私信问我怎么从基础学习，为了更加符合这些同学的基础，增加了最基础的知识内容，方便大家学习。

另，在最后增加了所有提到的书籍的全名，方便大家查找，有pdf链接的我会放上pdf的链接。如侵害了版权，请告知我。

删减内容：我把关于关于linux使用和运维的内容删去了，因为这本不是我擅长的内容，放在这里未免误人子弟，尤其是在杨周的提醒之下，才知道对运维的理解是错误的，所以觉得瞎写的话不免有悖知乎的精神，我想会有更多人会有更好的答案。

说说我的学习经历吧。因为我对这方面也非常感兴趣，而且年龄和背景和题主十分相似。我觉得要深入学习linux，有三个方面是要全面了解的。（下面有些东西会写成英文，因为有些东西的中文翻译我真的不太清楚，所以为了最大程度的避免误人子弟，我还是尽量用词准确，见谅）

1. 最重要的linux的内核工作方式：包括但不限于：

- 内存分页
- 进程切换
- interrupt 和 exception，以及system call
- lock
- 进程管理
- 内存管理和分配
- signal
- 文件系统（ext2，ext3）
- 输入输出设备
- 缓存

书可以看《Understanding the Linux Kernel, Third Edition [Daniel P. Bovet, Marco Cesati]》，

《IA-32 Intel® Architecture Software Developer's Manual Volume 3: System Programming Guide》

这个是理解linux的基础的基础，对于理解操作系统有莫大的好处。而且无论对于high level还是low level的编程都是相通的，很多东西在学完操作系统之后，你对他们的理解就完全不一样了。就像学好缓存在对于提高大运算上面的效率是很有帮助的。

一边看书一边可以写一个简单的linux自己玩，时长一个学期比较合适。因为课程要求，所以我写过可以多任务运行的linux，附带一个声卡driver。当时演示的时候放的是凤凰传奇，教授整个人斯巴达了。但是不得不说这是一个非常大的工作量。

但是无论是要做内核开发还是驱动开发，我觉得这些都是基础中的基础，是要扎扎实实掌握的，不得有半点马虎。

2. CPU架构/汇编

操作系统跟CPU的架构是离不开的，我个人觉得是相辅相成的。很明显的能看到Intel为了和操作系统配合起来，实现了很多硬件功能。最明显的就是虚拟内存已经可以在硬件层面上直接换算，不需要通过软件了。不懂x86或者arm的架构，会一定程度上造成理解linux内核的困难。比如为什么要以某种特定的方式分页，为什么内存管理的时候需要不同的算法，interrupt，system call的意义何在？等等等等的问题都需要精通CPU的架构。

当时就是因为对TLB的理解不够深入，不够仔细，造成我的系统每次切换任务的时候，会page fault。

重点在于interrupt的处理，缓存的概念和内存相关结构的设计。至于乱序执行，pipeline之类的都是次要的，对于linux不是特别的重要。

x86的汇编是基础，有助于debug和理解linux源代码。我写系统的时候大段写过x86代码来实现底层操作。不知道真正的linux是不是这样实现的。

3. high level编程

系统毕竟是为了应用程序而服务的，就像有需求才有生产一样。没有应用程序的需求，就不会设计相关的系统功能。了解high level的编程一定程度上可以融会贯通。

学习过程：以下均是为大一大二的朋友们写的，有更好建议的朋友请留言指正。

1. 逻辑电路，数字电路，简单处理器，汇编，C语言，参考书籍《Yale N. Patt and Sanjay J. Patel, *Introduction to Computing Systems: from bits and gates to C and beyond*, 2nd Edition, McGraw-Hill, 2003.》我强烈建议这本书，这也是学校的教科书。原因是有同学反映即使是计算机系的学生看底层的书仍然看不懂，这本书从最基本的逻辑电路讲起，再上升到汇编和机器语言，最后很自然的过渡到了C语言，所有这些内容都建立在一个简单的LC-3处理器的基础上，我相信即使是工作多年的朋友看这本书仍然会说“原来如此”。时长：一个学期，对于有基础的同学，半个学期。

附录中，有LC-3处理器的所有细节，包括电路，对处理器没有任何概念的同学是非常好的补充材料。

2. 数据结构，OOP，参考网站：cplusplus.com, ocw.mit.edu/courses/ele...

时长：一学期

3. x86汇编，[x86 Assembly](#)

从LC-3到x86[courses.engr.illinois.edu...](#)

4. 操作系统：参考最上。

《Understanding the Linux Kernel, Third Edition [Daniel P. Bovet, Marco Cesati]》idak.gop.edu.tr/esmeray...

《Yale N. Patt and Sanjay J. Patel, *Introduction to Computing Systems: from bits and gates to C and beyond*, 2nd Edition, McGraw-Hill, 2003.》链接没有，自己买吧

[Intel速 64 and IA-32 Architectures Software Developer Manuals](#)

编辑于 2014-06-04

▲ 657 ▼ 68 条评论 分享 收藏 感谢 ... 收起 ^

**Tanky Woo**Python/运维 at 知道创宇, tankywoo.com

84 人赞同了该回答

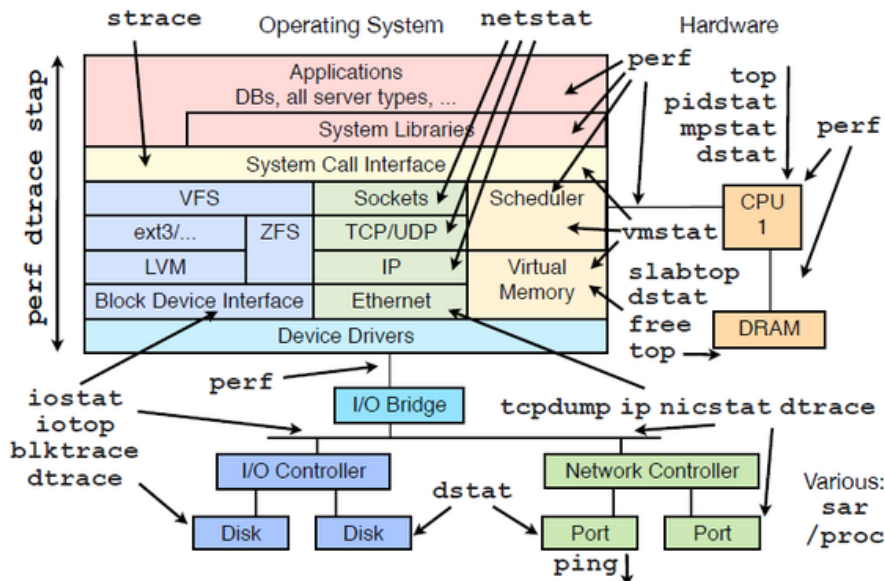
研究Linux内核是一个不错的选择

网络方面也是一个必备技能。

如果是运维，这个要求比较广：

最简单的就是会用命令，我不清楚你提到的熟悉各种命令到底是熟悉多少/熟悉到什么程度，除开日常使用的大部分命令，运维排查问题还需要了解很多命令，把这些工具用好，你可以解决很多问题。

借用网上的一幅图，看看你了解多少（反正我还有一些不了解 :（）：



运维的另外一方面就是根据需求，用好框架，比如Nagios, Puppet, SaltStack, Docker等，这方面的基础搭建要求非常高，你需要考虑如何做基础构建，才能达到高度模板化、自动化等。至少现在这方面都是我们Boss来做。

运维更多的就是和业务绑定在一起，根据业务来做开发和调整。

还有一方面，建议掌握一门除开Shell外的脚本语言，个人建议是Python。用合适的工具，做合适的事情，只靠Shell很难做好。

就说那么多了，如果你在大二就能达到你说的那个程度，那你已经很棒了，继续钻研吧。我觉得迷茫是正常，但是没必要去过多的想，随心去学，不是很好么。

编辑于 2014-05-01

▲ 84

● 14 条评论

➦ 分享

★ 收藏

♥ 感谢

...

收起 ^

**Bellevue** ★

编程 话题的优秀回答者

54 人赞同了该回答

1. 装 arch
2. 半年不更新
3. pacman -Syu
4. 把系统修好

编辑于 2016-01-01

▲ 54 ▼ ● 17 条评论 ↗ 分享 ★ 收藏 ♥ 感谢 ...



孙立伟

资深码农，技术死宅 ☹️

24 人赞同了该回答

给你一个网站，[Welcome to Linux From Scratch!](#)，尽情的折腾吧。

发布于 2014-04-29

▲ 24 ▼ ● 3 条评论 ↗ 分享 ★ 收藏 ♥ 感谢 ...



地球的外星人君

发现更多Linux知识：[t.cn/RogOHa1](#)

21 人赞同了该回答

先看题主的情况：

我是一个在校的大二学生，对 Linux 非常感兴趣，以后想往 Linux 方向发展

题主尚未毕业，有大把的时间用来研究Linux，同时自己有足够的热情和动力去持续学习。

但是这里有一个问题，题主并不明确自己未来的职业方向。表面上看题主是有一个远期规划的，但

实际上Linux方向有无穷多个分支，各个分支的技能要求差异会非常大。

我只是简单罗列了一下我脑子里的分支，并没有仔细查证：

a)Linux的系统、网络、服务、集群、网站、网络应用方向：

- 1、Web应用服务器，如sina、百度等大型网站
- 2、Mail应用服务器，如163或外企mail系统等
- 3、中间件或J2EE服务器，如为JBOSS Weblogic做平台
- 4、网络应用等。

b)嵌入式开发、UNIX/Linux应用系统开发，Linux内核驱动开发方向，主要有以下几类：

- 1、Linux下的C/C++ 系统程序开发
- 2、Linux平台Java体系开发和PHP开发
- 3、Linux下的图形界面开发
- 4、Linux底层内核/驱动开发
- 5、嵌入式Linux开发等。

c)Linux下的数据库，如Mysql、oracle和windows下的SQL Server及DB2等。

你看，现在就这么多，如果仔细查验一下会有更多。

所以，我第一条建议是，更深的了解一下自己的职业规划——做到细分比如DBA、web服务器当然不可能，但最少要有一个大的方向。

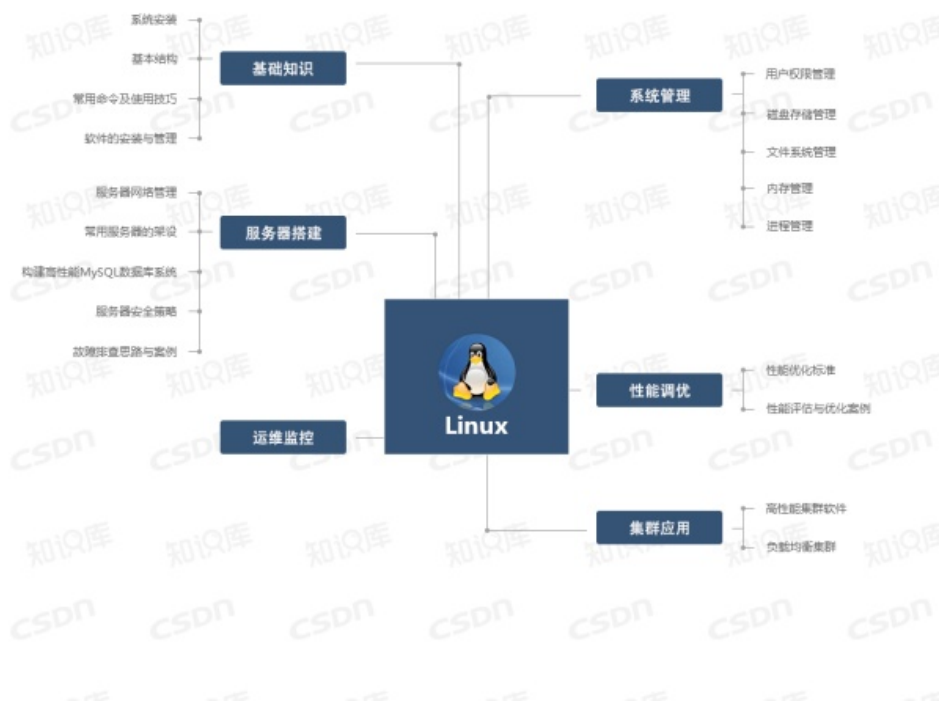
现在能熟练的操作 Linux 系统，基本上熟悉了各种命令，也能写一些 shell 脚本，平时写代码都用 Linux，基本上可以独立搭建各种开发环境，C 语言、操作系统原理学的不错

题主已经不算入门级的学习者了，最少在日常应用上应该已经相当熟练。作为一个自己学校未开设相关课程的大学生来说，自学能力相当强。

不知道如何才能更加深入的学习 Linux，比如 Linux 内核，驱动开发或者运维等等

出现这个疑问，其实和你没有明确自己的职业方向有很大关系。不过对于大二的学生来讲，确定职业方向几乎是一个不可能完成的任务。所以我就大致的推荐一下一些必备技能，其他的题主可以在自己学习的过程中一点一点的去接触。

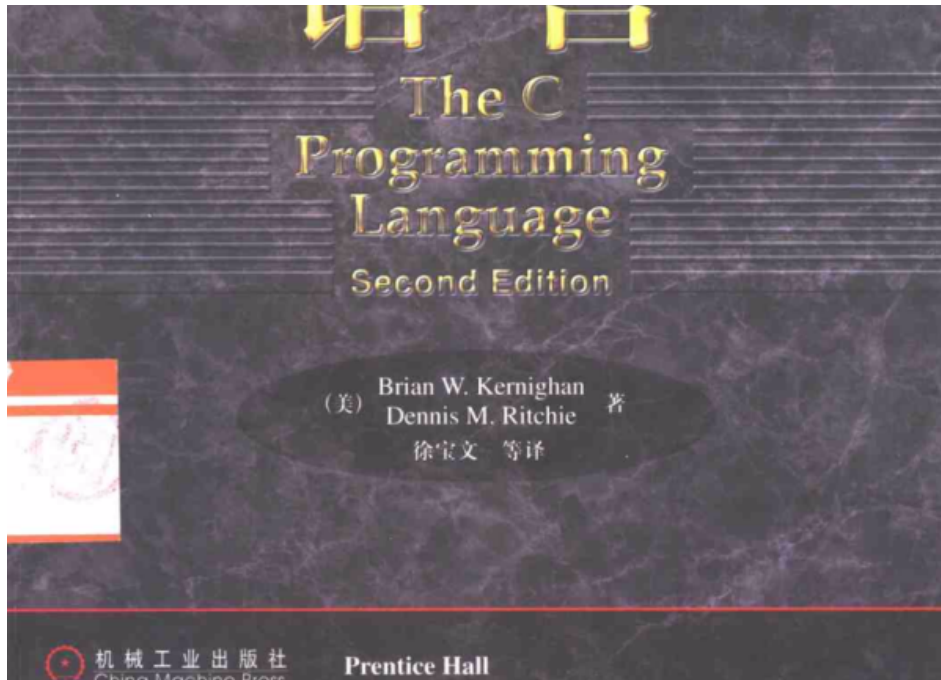
Linux知识结构图（来自CSDN）：



一些相关书籍的整理：

《The C Programming Language》





著名的计算机科学家Brian W. Kernighan和C语言之父的Dennis M. Ritchie合著的一部计算机科学著作，是第一部介绍C语言编程方法的书籍。它是一本必读的程序设计语言方面的参考书。它在C语言的发展和普及过程中起到了非常重要的作用，被视为是C语言的业界标准规范，而且至今仍然广泛使用。它也被公认为计算机技术著作的典范，以清晰简洁的文字讲述而著称。书中用"hello world"为实例开始讲解程序设计，也已经成为程序设计语言图书的传统。

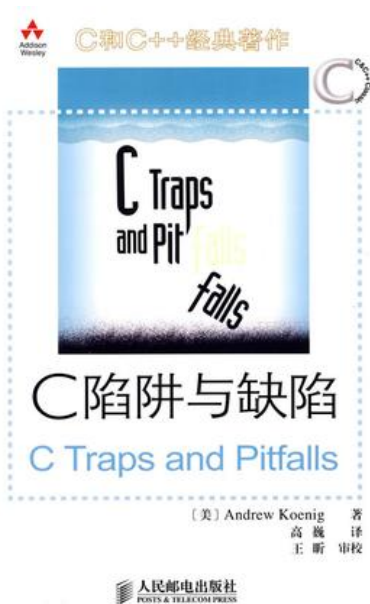
《C专家编程》



书中C的历史、语言特性、声明、数组、指针、链接、运行时、内存以及如何进一步学习C++等问题进行了细致的讲解和深入的分析。全书撷取几十个实例进行讲解，对C程序员具有非常高的实用价值。

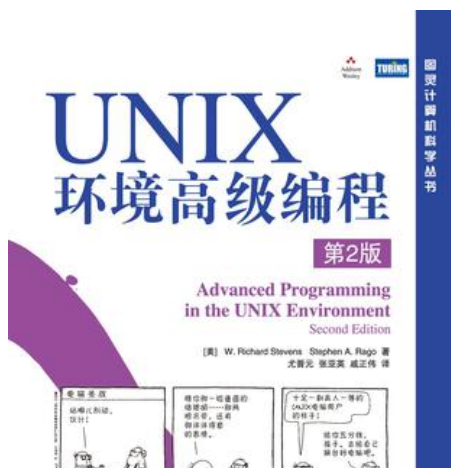
本书可以帮助有一定经验的C程序员成为C编程方面的专家，对于具备相当的C语言基础的程序员，本书可以帮助他们站在C的高度了解和学习C++。

《C陷阱与缺陷》



Andrew Koenig 是AT&T公司Shannon实验室大规模编程研究部门中的成员，同时也是C++标准委员会的项目编辑。他的编程经验超过30年，其中有15年在使用C++，他已经出版了超过150 篇和C++有关的论文，并且在世界范围内就这个主题进行过多次演讲。

《Unix环境高级编程》





W.Richard Stevens, 备受赞誉的技术作家, 生前著有多种经典的传世之作, 包括《UNIX网络编程》(两卷本)、《TCP/IP详解》(三卷本)和本书第1版。尤晋元, 上海交通大学计算机科学及工程系教授、博士生导师。在科研方面, 主要从事操作系统和分布对象计算技术方面的研究; 在教学方面, 长期承担操作系统及分布计算等课程的教学工作。主编和翻译了多本操作系统教材和参考书, 包括《UNIX操作系统教程》、《UNIX高级编程技术》、《UNIX环境高级编程》和《操作系统: 设计与实现》等。

《Unix网络编程》第二卷



《UNIX网络编程》(第1卷)(套接口API第3版)第1版和第2版由已故UNIX网络专家W. Richard Stevens 博士独自编写。《UNIX网络编程》(第1卷)(套接口API第3版)是3版, 由世界著名网络专家Bill Fenner 和Andrew M. Rudoff执笔, 根据近几年网络技术的发展, 对上一版进行全面修订, 增添了IPv6的更新过的信息、SCTP协议和密钥管理套接口的内容, 删除了X/Open传输接口的内容。

《用TCP/IP进行网际互连》第一卷



《用TCP/IP进行网际互连第1卷》是关于计算机网络的最著名的经典教材。它是目前美国大多数大学所开设的计算机网络课程的主要参考书。目前国内外能见到的各种关于TCP/IP的书籍，其主要内容都参考了《用TCP/IP进行网际互连第1卷:原理、协议与结构(第5版)(英文版)》。《用TCP/IP进行网际互连第1卷:原理、协议与结构(第5版)(英文版)》的特点是：强调原理、概念准确、深入浅出、内容丰富新颖。各章后附有很好的习题。这一卷是2006年出版的第五版，分为四个部分。第一部分为概述，第二部分从单个主要来看TCP/IP互联网，第三部分则是从全局来看互联网的结构，第四部分讨论因特网提供的应用层服务。

《Linux程序设计》





Neil Matthew 和Richard Stones世界知名的Linux/UNIX专家，有数十年Linux/UNIX开发经验和从业经历。他们使用过几乎所有UNIX版本，并精通C/C++、LISP、Fortran、Perl、Tcl和Prolog等各种语言。他们从事过各种软件项目，从实时嵌入式系统到会计系统和零售信息系统。除本书外，他们还合著了PostgreSQL、MySQL方面的图书。

《Linux DeviceDriver》

相关问题

Linux 为什么还要坚持使用宏
个回答

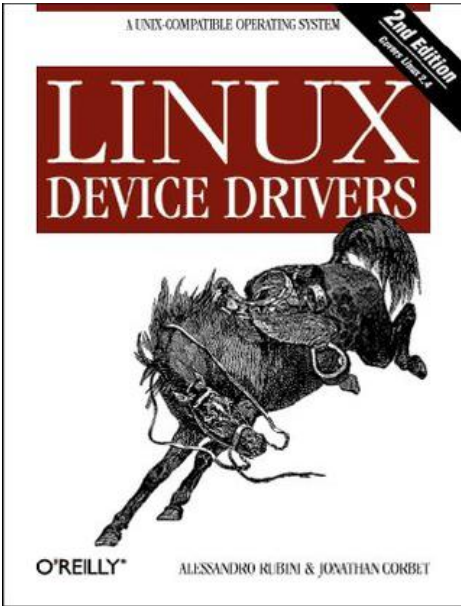
用 Linux 真的能学到很多平台
吗？ 91 个回答

Windows NT 内核与 Linux 的
更安全？ 15 个回答

UNIX 文化中有哪些元素？ 2

关注问题

如何更深入地学习 Linux?



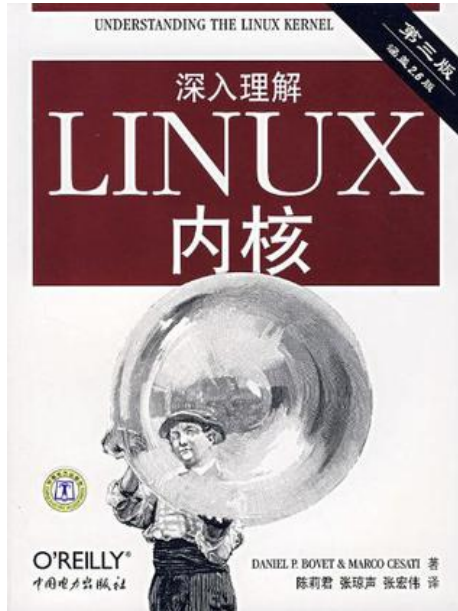
Jonahan Corbet早在1981年就开始接触BSD Unix的源代码。那时，科罗拉多大学的一名讲师让他“修正”其中的分页算法。从那时起直到现在。他深入研究了他所遇到的每一个系统，其中包括VAX、Sun、Arden以及x86系统的驱动程序。他在1993年第一次接触Linux系统，从此以后一直从事Linux的开发。Corbet先生是《Linux Weekly News》的创始人和执行主编。他和妻子及两个孩子生活在科罗拉多州的波尔得市。

《深入理解Linux内核》

相关 Live 推荐

- Web 开发那些事儿
- 如何快速掌握新技术
- 安卓入门和求职指南
- 前端工程化浅析
- 车载控制软件设计：产

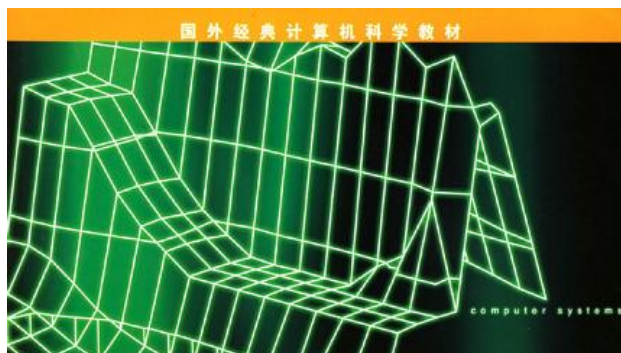
刘看山 · 知乎指南 · 知乎协议 · 应
侵权举报 · 网上有害信息举报专区
违法和不良信息举报：010-827166
联系我们 © 2017 知乎



博韦，计算机专业博士，意大利罗马大学Tor vergata分校全职教授。

切萨蒂 数学和计算机科学博士，罗马大学Tor vergata分校工程学院计算机科学系助理研究员。

《深入理解计算机系统》





本书适用于那些想要写出更快、更可靠程序的程序员。通过掌握程序是如何映射到系统上，以及程序是如何执行的，读者能够更好的理解程序的行为为什么是这样的，以及效率低下是如何造成的。粗略来看，计算机系统包括处理器和存储器硬件、编译器、操作系统和网络互连环境。而通过程序员的视角，读者可以清晰地明白学习计算机系统的内部工作原理会对他们今后作为计算机科学研究者和工程师的工作有进一步的帮助。它还有助于为进一步学习计算机体系结构、操作系统、编译器和网络互连做好准备。

这些内容足够一个不确定自己职业方向的Linux学习者用来自我提升了，如果你确定了职业方向，就应该更加针对性的学了。

发布于 2017-05-09

21 1 条评论 分享 收藏 感谢 ... 收起



汤哥

Linux两千人新人学习交流群：576095618

5 人赞同了该回答

成为一名精通 Linux程序设计的高级程序员一直是不少朋友孜孜以求的目标。

根据中华英才网统计数据，北京地区 Linux 程序员月薪平均为 Windows程序员的 1.8 倍、Java 程序员的 2.6 倍，Linux 程序员年终奖金平均为 Windows 程序员的 2.9倍。同时数据显示，随着工作经验的增长，Linux 程序员与 Windows 程序员的收入差距还有扩大的趋势。

Denis 认为，要学好Linux 环境下的编程，关键是要看对、选对、学会正确的书籍。可以说，如果你选对了Linux编程的经典书籍，配合你在程序设计工作中的刻苦钻研，成为一名精通 Linux程序设计的高级程序员并非一件可望不可及的事情。但如果各位程序员朋友没有选对正确的书籍，则你的职业生涯之路就可能面临坎坷。

今天，汤哥向各位朋友推荐的这些书，有的是资深老前辈们当初向汤哥的推荐，还有的是IBM 的内部培训指定参考书，它们都很值得各位朋友抽空认真一读。

为什么要学习 Linux 环境下的编程

Linux 是一个开放、灵活、跨平台的操作系统，上至庞大的数据中心，下至可放于掌心中的嵌入式设备，无处没有 Linux的身影。更为重要的是，Linux 是一个与 Unix 一脉相承又与时俱进的系统。可以说，上世纪70年代学习的 Unix知识和技巧，在今天仍然大有用武之地，这与 Windows 平台的开发形成了鲜明的对比。程序员不用担心今天微软出一个.net，明天又出一个 F#，使得自己过去学习的成果付之东流

上个星期，汤哥与一位 Linux 项目经理聊天过程中，这位 Linux 项目经理告诉汤哥，他们项目的利润非常高，急需具备一定Linux 编程知识的程序员。他说：“其实我们对程序员的编程技巧要求也并非很高，这是可以在工作中培训和提高的，关键是很多程序员连基本的Linux 编程思想都不了解，我怎么聘用他们呢？我们去大学招聘的时候，给本科应届生开出 8000元的月薪，但是就是很难招到人。我拿一些经典书籍中一些经典例子来考他们，他们基本上都是一问三不知。其实，如果

他们能回答上一半的问题，我还是很愿意考虑是否聘用他们的。而对于项目相关部分的专业知识，我们有专业的内部培训，并不担心这个问题，关键是看应聘者是否具备Linux 编程的基本思想。”汤哥认为，这位项目经理朋友提到的问题还是很有代表性的。其实很多程序员朋友，只要能掌握这些书中的基础知识，是很容易脱颖而出的。事实上，项目经理他们也都很忙，并没有空去自己设计高难度的面试题，而是直接采用经典书籍中的经典例子。

《UNIX环境高级编程》（第2版），史蒂文斯著

《UNIX 环境高级编程》，推荐指数：★★★★★

《UNIX环境高级编程》是 Unix/ Linux 程序员案头必备的一本书籍。可以说，Linux程序员如果没有读过这本书，就好像基督教徒没有读过圣经一样，是很难让人理解的。这本书概括了 Linux 编程所需的一切理论框架、主要系统函数、多进程编程、乃至 Linux网络通信。对于初学者，如果你能将《操作系统》这门课程结合着这本书来学习，试着用 Linux程序实践《操作系统》这门书中讲的进程间通信、进程调度、进程同步等内容，相信这将是一个一举两得的事情。哦，忘了告诉大家，上次 CNN采访 Google 总裁佩林的时候，汤哥看到佩林的书架上就有这本书的英文版，可见它真的是一本全球开发者必备的一本书。

《深入理解 Linux 内核》（第三版），博韦等著

《深入理解 Linux 内核》，推荐指数：★★★★★

学习 Linux，就要学习 Linux 的精华。而 Linux 的精华，则在于 Linux 的内核。《深入理解 Linux内核》就是一本辅助学习 Linux 内核的经典书籍。有的初学者，在没有人指导的情况下，就钻入 LinuxKernel代码的海洋中埋头苦学，结果学了半天仍然是一头雾水。当然了，在大师指导下学习就不一样了。本书以最新的 Linux 2.6版架构为基础，分门别类地向初学者介绍了 Linux 内核的架构、编程思想、以及功能模块。相信你在本书的指导下学习，对于你读懂Linux 操作系统的精华部分，会取得事半功倍的效果。事实上，不少知名公司招聘的题目里面，很多就出自这本书，可见它真的是论述Linux 内核的经典书籍。

《Linux 设备驱动程序》，科波特著

《Linux 设备驱动程序》，推荐指数：★★★

程序开发，高薪在 Linux；Linux 程序员，高薪在驱动开发。可以说，汤哥见过的 Linux驱动程序，薪水在万元以下的，一个都没有。普遍观点认为，Linux 驱动开发很难学，这并不正确。初学者认为 Linux驱动开发很难学，关键在于其没有选对正确的入门书籍。《Linux 设备驱动程序》这本书，过去已经有多位 Linux项目经理向汤哥推荐过，你要想学MOD编程，想挑战高薪职位，这本书读一读是很有必要的。当然，这本书不是一本针对初学者的书籍，初学者应首先学习前面提到的《UNIX环境高级编程》再看这本书，你才能真正的学懂学好Linux 驱动开发。

《嵌入式 Linux 应用开发完全手册》，韦东山著

《嵌入式 Linux 应用开发完全手册》，推荐指数：★★★★

要说 Linux 应用最广泛的地方，那一定是无处不在的嵌入式设备了，《嵌入式 Linux 应用开发完全手册》就是一本教会您怎样开发Linux嵌入式系统软件的一本好书。汤哥与本书作者韦东山老师曾有过一面之缘，当时韦老师正在写作这本书。这本书综合了常见的嵌入式开发经验技巧，以及常见的嵌入式系统应用，系统移植，调试及异常处理等，内容非常丰富，是中文Linux领域难得的一本好书。嵌入式处理器种类繁多，韦老师着重讲了目前国内最常用的ARM系统，实用性很强，是国内嵌入式程序员不容错过的一本工具书。

《人月神话》（32周年中文纪念版），布鲁克斯著

《人月神话》，推荐指数：★★★★★

汤哥上面为大家推荐了几本 Linux编程经典书籍，但要问编程的最终目的是什么，那一定还是要应用到实际项目中。做项目，那就一定离不开《人月神话》这本书。《人月神话》的作者布鲁克斯，是汤哥所在公司的超级大牛人，也是图灵奖的获得者。去年，他到中国访问的时候曾经表示，《人月神话》这本书，浓缩了项目实践中的正反两方面经验，是项目经理和系统分析师必读的一本书。汤哥认为，各位年轻的朋友，最终还是得要管项目的，与其你临时再学习，还不如早点接触这方面的知识比较好。可以说，Linux的项目，对开发者的要求都很高，大家不要去跟 Windows程序员学一些不好的习惯，特别是项目实施方面的一些不好的习惯，而应该深入体会大师的书籍。目前，

《人月神话》的32周年纪念版已经上市了，有志于成为项目经理和产品经理的人，不应该错过这本做项目的圣经。

总结

古人云：“生而知之者，上也；学而知之者，次也；困而学之，又其次也；困而不学，民斯为下矣。”汤哥认为，正如古人所言，对于成功的Linux 程序员，勤奋苦读的结果，将为您的 Linux 学习之路锦上添花。对于还没有入门的投资者，尽快熟读一两本入门的 Linux书籍，对于你少走弯路是很重要的。对于已经进入项目，但还不能熟练掌握 Linux开发的程序员，越早补习自己缺乏的东西，您将越早受益。对于基础薄弱，又拒绝学习的程序员，或许“民斯为下矣”就将是您惨淡的结局。

编辑于 2017-05-10



赖明星

10 人赞同了该回答

好巧，我最近正在看

linux内核这么高深的知识，居然读起来不累，推荐。

等我以后有能力了，也要写一本这样的书。

编辑于 2017-01-26

▲ 10 ▼ ● 添加评论 ➦ 分享 ★ 收藏 ♥ 感谢 ... 收起 ^



chinatree

從事運維，關注信息安全。

8 人赞同了该回答

一谈到深入学习Linux，多数人马上联想到Linux内核，但这是一个较长远目标，建议这个定位在**重要但不紧急**的任务。因为如果内核就接触就能接触的话，那这就不可以算啥深入的知识啦。

至于楼主说的熟悉各种命令，能写Shell脚本，不确定是到什么程序，建议楼主寒暑假有时间的话可以去找一两家公司尝试面试一下验证自己有哪些不足 :)

楼主现在是一个在校大二学生，迷茫是正常现象，之所有产生迷茫主要原因在于没有真实案例给你展示你的能力，而你又觉得你所要学的基本都掌握得差不多。然而人是渴望得到认可、得到掌声的，现在你的能力暂时还没有发挥用武之地，所以缺少了认可和掌声提供的动力，以后你在实际工作中所完成的每一项任务给你带来的成就感和满足感，以及你发现比你厉害的牛人还很努力的学习的时候，你就会清晰的知道自己该往哪个方向加油。

Linux内核和驱动开发我不熟悉，没啥可建议的 :(

运维(ops)职业，要求的知识面很广泛，除了Linux知识，开发的一些技术、网络知识、脚本能力、排错能力、安全等等，像 @Tanky Woo 的图片中所涉及到的命令、知识点在监控、排障能力上体现得淋漓尽致。

最后，学校的知识对实际工作来说，那真的只是冰山一角，提醒楼主多把自己所学知识付诸实践，按老马(马克思)说的，实践是检验真理的唯一标准。在实践中学会思考问题，提出问题，解决问题，这样你会发现要学的东西很多很多，简单的可以理解为属于你现在不懂的知识都可以算作深入学习...

发布于 2014-05-08

▲ 8 ▼ ● 1 条评论 ➦ 分享 ★ 收藏 ♥ 感谢 ...



Wayne Shi

Learn by doing

22 人赞同了该回答

Linux内核可以跟着中科大孟宁老师的MOOC学 [Linux内核分析](#)，孟老师讲课很有意思，把枯燥难以理解的操作系统知识讲的很透彻，对理解Linux操作系统运行的基本过程及涉及的核心机制很有帮助。

实验楼为这门MOOC提供了配套的在线linux实验环境，理论结合实践才能抓住Linux内核的精髓。

[Linux - Linux内核分析](#)

编辑于 2016-02-25

▲ 22 ▼ ● 添加评论 ➦ 分享 ★ 收藏 ♥ 感谢 ...



Victor King

一个徘徊在理想和现实之间的Loser

如何才算是深入？Linux的面太广了。我觉得如果迷茫，那就想想“学以致用”这个词。你想想你将来打算用Linux 做什么。

如果不知道，我说几条，仅供参考。

- 1、问问你的老师，最好是带研究生的那种，看看他们的看法，至少对你以后继续往上考有帮助
- 2、把你平时用的东西变成Linux的，比如服务器，比如软件开发，比如路由器
- 3、参加比赛，尤其是有Linux的那种比赛
- 4、买点大部头的书开始啃，比如Linux程序开发，内核分析之类
- 5、找实习机会——但我也不太赞同这个法子，毕竟才上大二，后面还有很多东西要学，也许你会发现，原来我学的东西已经可以应付现在的工作了。这对你搭建一个知识系统没什么好处

发布于 2016-08-16

▲ 0 ▼ ● 添加评论 ➦ 分享 ★ 收藏 ♥ 感谢 ...



binape

全栈/强势闷骚/美腿控

1 人赞同了该回答

oldlinux.org

发布于 2016-02-25

▲ 1 ▼ ● 2 条评论 ➦ 分享 ★ 收藏 ♥ 感谢 ...



夏华林

5 人赞同了该回答

推荐一本书，《深入Linux内核架构》

有精力的话，可以读一读，书挺厚，中文版翻译的也不错。

发布于 2014-04-29

▲ 5 ▼ ● 3 条评论 ➦ 分享 ★ 收藏 ♥ 感谢 ...



weez1024

政治不正确。

3 人赞同了该回答

《30天自制操作系统》

先从一个小的file system做起。

编辑于 2014-06-04

▲ 3 ▼ ● 4 条评论 ➦ 分享 ★ 收藏 ♥ 感谢 ...



Name5566

1 人赞同了该回答

建议你先确定发展的方向（还是做驱动？还是做网络？等），Linux 包含内容太多，确定方向才能选择合适的点深入下去。找到需要深入的点，就选择相关的书籍学习一下，自己编写一些程序实践，在实践过程中回顾存在知识点的不足，再进一步学习。同时可以关注领域内的动态。

发布于 2014-05-01

▲ 1 ▼ 1 条评论 分享 收藏 感谢 ...

**李超**

data mining on Spark & Hadoop

2 人赞同了该回答

俺不是科班出身，所以给你提供一点新鲜的见解，Linux是一个让我玩得很happy的系统，比如怎样用shell+tmux扔掉鼠标，怎样把emacs打造成全能环境，怎么最酷地用vim，怎样安装Arch Linux，比较各种桌面环境的优缺点等等，总之鼓捣Linux是满足人好奇心的，如果你现在拿着Linux不知道学什么，就把心态放轻松，把它当玩具来玩好了。

发布于 2014-06-03

▲ 2 ▼ 6 条评论 分享 收藏 感谢 ...

**酿泉**

懒得删空答案，请当作已注销

4 人赞同了该回答

用 ArchLinux 就行了

发布于 2014-05-01

▲ 4 ▼ 6 条评论 分享 收藏 感谢 ...

**Roladrw**

不要咬我！我只是想逗你一下！

12 人赞同了该回答

从卸载Windows开始。

发布于 2014-06-04

▲ 12 ▼ 3 条评论 分享 收藏 感谢 ...

**枕水**

C语言

2 人赞同了该回答

如果题主所说属实，那应该是已经到了一个相当的境界了，就这种程度将来求职就不成问题了，我同学当年在学校的时候就一直在玩Linux，现在想想也就是到了会配置Linux的程度，差不多我们装Linux都要找他过来这样子。然后毕业就有公司要，据说工资差不多5K的样子，在我印象里05年毕业生拿5k工资还是不少的吧？

我是工作之后才接触Linux的，感觉光是“熟练的操作 Linux 系统，熟悉各种命令，独立搭建gcc编译链（而不是找现成的改改参数这种）”就已经算人才了。我不知道题主说的写脚本是不是就是指的写make文件。

要再深入的话，除了读懂Linux源代码之外，是不是可以加入Linux的邮件列表里？试着跟一下开发进度？最不济也混个脸熟，看看大牛们是怎么讨论代码的。Linux的种种bug真是恨死人，你不知道bug的话，经常代码调了半天不通过，还以为是自己的问题。

记得我以前公司的一个大牛，跟人讨论起Linux的一个系统功能的时候，直接调出源代码讲Linux在这块是怎么实现的，那逼格真是满到爆表啊！

深入玩Linux的话，就不要搞运维了，感觉好浪费的样子，虽然有时候开发的代码水平还比不上运维的。

发布于 2014-06-02

▲ 2 ▼ 2 条评论 分享 收藏 感谢 ...

**Tiger David**

 哭比博士后

4 人赞同了该回答

运维的话，我不太清楚，不过总觉得在没有实际服务器的情况下学习运维，就好像在没有龙的世界学习屠龙剑法一样，学了再多的知识也没有实感阿。。。

而且楼主大二就对Linux如此有兴趣，要是以后工作就干运维的话有点浪费了

我个人觉得工具这东西，只有在有实际需求的情况下，学习才有动力和意义。题主说到了几个方向，比如说学习linux内核，驱动开发或者运维。这些在没有实际目的的情况下，很容易就半途而废得，因为你说的这些方向都不是简单的活儿。。。

我就举一个例子吧，想当年我开始深入研究Linux，就是因为那会儿我被人忽悠入了一个HP Touchpad。。。现在想想满眼都是泪阿，WebOS系统虽然好看但基本没啥软件支持得，Flash都不支持所以我Bilibili弹幕网站什么都上不了。。。所以当年我就开始研究怎么刷Android系统。而Android系统编译Rom最好的环境就是Linux，我也因此开始折腾之旅了：先是用别人的代码，然后编译不通过，然后尝试Debug，然后为了能看懂代码开始熟悉Android系统框架，然后开始熟悉Android APP编程。。。虽然这一切在我入了iPad后就中止了，不过也因为这个，让我学到不少知识了吧。。。

所以我觉得题主可以给自己设定一个project，比如自己实现一个小操作系统之类，然后以这个为目的开始学习，效率啊乐趣之类都能够提高不少，更不用说学习的目的性了。

发布于 2014-05-05

 4  11 条评论  分享  收藏  感谢 ...全能手
计算机行业

3 人赞同了该回答

Linux这块学习其实分为几个阶段：

- 1、打基础:先将Linux基本的命令掌握
- 2、过度阶段:如果将来决定从事纯Linux运维，那么大部分工作岗位都是互联网提供的，所以要掌握Linux下的一些服务相关的内容。如果是从事数据库方面的工作，服务的东西了解架构即可，当然掌握最好咯。
- 3、提升阶段，可以拆分多个方向，当然在某些情况下又是相辅相成的。如果你对Linux内核开发有兴趣的话，那么你至少要掌握C或者C++等编程语言的使用，以及Linux的内核的设计原理，当然这个过程可能是漫长的，需要大量时间成本，也是可能赚钱最多的；如果你比较追求新鲜的概念，比如说到“云”这个东西，那么就要求你要掌握虚拟化技术，比如KVM、VMWARE、Docker等，以及行业内最流行的云架构，比如Openstack等，另外至少要掌握Linux shell、python等常用的脚本语言，python现在很火。
- 4、飞升阶段：就之前在过度阶段提到的数据库。掌握好现在乃至未来比较主流的数据库产品，比如：Oracle、MySQL、mongoDB等，会是你的人生轨道上升到一个新的层次。


最后，祝所有在技术岗位上打拼的人们，都会获得自己满意的工作。

发布于 2017-02-16

 3  1 条评论  分享  收藏  感谢 ...

更多

7 个回答被折叠（为什么？）

张春生
 编辑话题经验

使用匿名身份回答

写回答...