Linux 学习路线指南

作者：lxlcs201
原文链接：https://ld246.com/article/1534423276427
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

很多同学接触 Linux 不多，对 Linux 平台的开发更是一无所知。而现在的趋势越来越表明，作为个优秀的软件开发人员，或计算机 IT 行业从业人员，掌握 Linux 是一种很重要的谋生资源与手段。来我将会结合自己的几年的个人开发经验，及对 Linux，更是类 UNIX 系统，及开源软件文化，谈谈 Linux 的学习方法与学习中应该注意的一些事。
就如同刚才说的，很多同学以前可能连 Linux 是什么都不知道，对 UNIX 更是一无所知。所以我从最基础的讲起，对于 Linux 及 UNIX 的历史我们不做多谈，直接进入入门的学习。
Linux 入门是很简单的，问题是你是否有耐心，是否爱折腾，是否不排斥重装一类的大修。没折可以说是学不好 Linux 的，鸟哥说过，要真正了解 Linux 的分区机制，对 LVM 使用相当熟练，没有 0 次以上的 Linux 装机经验是积累不起来的，所以一定不要怕折腾。
由于大家之前都使用 Windows，所以我也尽可能照顾这些“菜鸟”。我的推荐，如果你第一次触 Linux，那么首先在虚拟机中尝试它。虚拟机我推荐 Virtual Box，我并不主张使用 VM，原因是 V 是闭源的，并且是收费的，我不希望推动盗版。当然如果你的 Money 足够多，可以尝试 VM，但我说的是即使是 VM，不一定就一定好。付费的软件不一定好。首先，Virtual Box 很小巧，Windows 台下安装包在 80MB 左右，而 VM 动辄 600MB，虽然功能强大，但资源消耗也多，何况你的需求 Virual Box 完全能够满足。所以，还是自己选。如何使用虚拟机，是你的事，这个我不教你，因为很简，不会的话 Google 或 Baidu 都可以，英文好的可以直接看官方文档。
现在介绍 Linux 发行版的知识。正如你所见，Linux 发行版并非 Linux，Linux 仅是指操作系统内核，作为科班出生的你不要让我解释，我也没时间。我推荐的发行版如下：

UBUNTU 适合纯菜鸟，追求稳定的官方支持，对系统稳定性要求较弱，喜欢最新应用，相对来不太喜欢折腾的开发者。
Debian，相对 UBUNTU 难很多的发行版，突出特点是稳定与容易使用的包管理系统，缺点是企支持不足，为社区开发驱动。
Arch，追逐时尚的开发者的首选，优点是包更新相当快，无缝升级，一次安装基本可以一直运作去，没有如 UBUNTU 那样的版本概念，说的专业点叫滚动升级，保持你的系统一定是最新的。缺点然易见，不稳定。同时安装配置相对 Debian 再麻烦点。
Gentoo，相对 Arch 再难点，考验使用者的综合水平，从系统安装到微调，内核编译都亲历亲，是高手及黑客显示自己技术手段，按需配置符合自己要求的系统的首选。
Slackware 与 Gentoo 类似。
CentOS，社区维护的 RedHat 的复刻版本，完全使用 RedHat 的源码重新编译生成，与 RedHat 的兼容性在理论上来说是最好的。如果你专注于 Linux 服务器，如网络管理，架站，那么 CentOS 是的选择。
LFS，终极黑客显摆工具，完全从源代码安装，编译系统。安装前你得到的只有一份文档，你要的就是照文档你的说明，一步步，一条条命令，一个个软件包的去构建你的 Linux，完全由你自己控，想要什么就是什么。如果你做出了 LFS，证明你的 Linux 功底已经相当不错，如果你能拿 LFS 文档学活用，再将 Linux 从源代码开始移植到嵌入式系统，我敢说中国的企业你可以混的很好。

你得挑一个适合你的系统，然后在虚拟机安装它，开始使用它。如果你想快速学会 Linux，我有个建议就是忘记图形界面，不要想图形界面能不能提供你问题的答案，而是满世界的去找，去问，如用命令行解决你的问题。在这个过程中，你最好能将 Linux 的命令掌握的不错，起码常用的命令得知，同时建立了自己的知识库，里面是你积累的各项知识。
再下个阶段，你需要学习的是 Linux 平台的 C/C++ 开发，同时还有 Bash 脚本编程，如果你对 Jva 兴趣很深还有 Java。同样，建议你抛弃掉图形界面的 IDE，从 VIM 开始，为什么是 VIM，而不是macs，我无意挑起编辑器大战，但我觉得 VIM 适合初学者，适合手比较笨，脑袋比较慢的开发者。macs 的键位太多，太复杂，我很畏惧。然后是 GCC，Make，Eclipse（Java，C++ 或者）。虽然将 C++ 列在了 Eclipse 中，但我并不推荐用 IDE 开发 C++，因为这不是 Linux 的文化，容易让你忽略些你应该注意的问题。IDE 让你变懒，懒得跟猪一样。如果你对程序调试，测试工作很感兴趣，GDB 得学的很好，如果不是 GDB 也是必修课。这是开发的第一步，注意我并没有提过一句 Linux 系统 API 的内容，这个阶段也不要关心这个。你要做的就是积累经验，在 Linux 平台的开发经验。我推荐的书下：C 语言程序设计，谭浩强的也可以。C 语言，白皮书当然更好。C++ 推荐 C++ Primer Plus，Ja a 我不喜欢，就不推荐了。工具方面推荐 VIM 的官方手册，GCC 中文文档，GDB 中文文档，GNU 源软件开发指导（电子书），汇编语言程序设计（让你对库，链接，内嵌汇编，编译器优化选项有初了解，不必深度）。
如果你这个阶段过不了就不必往下做了，这是底线，最基础的基础，否则离开，不要霍霍 Linux

发。不专业的 Linux 开发者作出的程序是与 Linux 文化或 UNIX 文化相背的，程序是走不远的，不可像 Bash，VIM 这些神品一样。所以做不好干脆离开。
接下来进入 Linux 系统编程，不二选择，APUE，UNIX 环境高级编程，一遍一遍的看，看 10 都嫌少，如果你可以在大学将这本书翻烂，里面的内容都实践过，有作品，你口头表达能力够强，你以在面试时说服所有的考官。（可能有点夸张，但 APUE 绝对是圣经一般的读物，即使是 Windows 序员也从其中汲取养分，Google 创始人的案头书籍，扎尔伯克的床头读物。）
这本书看完后你会对 Linux 系统编程有相当的了解，知道 Linux 与 Windows 平台间开发的差异哪？它们的优缺点在哪？我的总结如下：做 Windows 平台开发，很苦，微软的系统 API 总在扩容，使用最新潮，最高效的功能，最适合当前流行系统的功能你必须时刻学习。Linux 不是，Linux 系统核心 API 就 100 来个，记忆力好完全可以背下来。而且经久不变，为什么不变，因为要同 UNIX 兼，符合 POSIX 标准。所以 Linux 平台的开发大多是专注于底层的或服务器编程。这是其优点，当然形是 Linux 的软肋，但我站在一个开发者的角度，我无所谓，因为命令行我也可以适应，如果有更好图形界面我就当作恩赐吧。另外，Windows 闭源，系统做了什么你更本不知道，永远被微软牵着鼻跑，想想如果微软说 Win8 不支持 QQ，那腾讯不得哭死。而 Linux 完全开源，你不喜欢，可以自己，只要你技术够。另外，Windows 虽然使用的人多，但使用场合单一，专注与桌面。而 Linux 在各方面都有发展，尤其在云计算，服务器软件，嵌入式领域，企业级应用上有广大前景，而且兼容性一，由于支持 POSIX 可以无缝的运行在 UNIX 系统之上，不管是苹果的 Mac 还是 IBM 的 AS400 系列都是完全支持的。另外，Linux 的开发环境支持也绝对是一流的，不管是 C/C++，Java，Bash，Python，PHP，Javascript，。。。。。。就连 C#也支持。而微软除 Visual Stdio 套件以外，都不怎么好，不是吗？
如果你看完 APUE 的感触有很多，希望验证你的某些想法或经验，推荐 UNIX 程序设计艺术，界顶级黑客将同你分享他的看法。
现在是时候做分流了。 大体上我分为四个方向：网络，图形，嵌入式，设备驱动。
如果选择网络，再细分，我对其他的不是他熟悉，只说服务器软件编写及高性能的并发程序编写。相对来说这是网络编程中技术含量最高的，也是底层的。需要很多的经验，看很多的书，做很多的目。
我的看法是以下面的顺序来看书：

APUE 再深读 – 尤其是进程，线程，IPC，套接字
多核程序设计 - Pthread 一定得吃透了，你很 NB
UNIX 网络编程 – 卷一，卷二
TCP/IP 网络详解 – 卷一 再看上面两本书时就该看了
5.TCP/IP 网络详解 – 卷二 我觉得看到卷二就差不多了，当然卷三看了更好，努力，争取看了</li

6.Lighttpd 源代码 - 这个服务器也很有名了
7.Nginx 源代码 – 相较于 Apache，Nginx 的源码较少，如果能看个大致，很 NB。看源代码主是要学习里面的套接字编程及并发控制，想想都激动。如果你有这些本事，可以试着往暴雪投简历，他们写服务器后台，想一想全球的魔兽都运行在你的服务器软件上。
Linux 内核 TCP/IP 协议栈 – 深入了解 TCP/IP 的实现

如果你还喜欢驱动程序设计，可以看看更底层的协议，如链路层的，写什么路由器，网卡，网络备的驱动及嵌入式系统软件应该也不成问题了。
当然一般的网络公司，就算百度级别的也该毫不犹豫的雇用你。只是看后面这些书需要时间与经，所以 35 岁以前办到吧！跳槽到给你未来的地方！
图形方向，我觉得图形方向也是很有前途的，以下几个方面。

Opengl 的工业及游戏开发，国外较成熟。
影视动画特效，如皮克斯，也是国外较成熟。
GPU 计算技术，可以应用在浏览器网页渲染上，GPU 计算资源利用上，由于开源的原因，有很的文档程序可以参考。如果能进火狐开发，或 google 做浏览器开发，应该会很好 。

嵌入式方向：嵌入式方向没说的，Linux 很重要。
掌握多个架构，不仅 X86 的，ARM 的，单片机什么的也必须得懂。硬件不懂我预见你会死在半上，我也想走嵌入式方向，但我觉得就学校教授嵌入式的方法，我连学电子的那帮学生都竞争不过。

劝大家，一定得懂硬件再去做，如果走到嵌入式应用开发，只能祝你好运，不要碰上像 Nokia，Hp 样的公司，否则你会很惨的。
驱动程序设计：软件开发周期是很长的，硬件不同，很快。每个月诞生那么多的新硬件，如何让们在 Linux 上工作起来，这是你的工作。由于 Linux 的兼容性很好，如果不是太低层的驱动，基本 C 语言就可以搞定，系统架构的影响不大，因为有系统支持，你可能做些许更改就可以在 ARM 上使用 C 的硬件了，所以做硬件驱动开发不像嵌入式，对硬件知识的要求很高。可以从事的方向也很多，如电啊，特别是如索尼，日立，希捷，富士康这样的厂子，很稀缺的。

LDD – Linux 驱动程序设计与内核编程的基础读物
深入理解 Linux 内核 – 进阶的
Linux 源代码 – 永无止境的

当然你还的看个方面的书，如网络啊什么的。