

0. 跟着项目走；

1. 学会用、熟练用linux系统；

2. Linux Kernel Development.

3. Understanding the Linux Kernel.

4. Linux Device Driver.

5. 找一个感兴趣的子系统研究，认真研究代码，这里的建议是可以看提交的每个change。比如网络子系统：Understanding linux network internals.

“严谨而细致的仔细分析它的代码，不懂的地方就通过社区、邮件列表或者直接发Email给 maintainer请教等途径弄懂，切勿得过且过”。

“通过Kconfig与Makefile定位目标代码”。

6. 融入社区做贡献.....

坚持！坚持！！坚持！！！

1. 了解操作系统基本概念。如果不会，可以学习《操作系统：设计与实现》

Andrew S.Tanenbaum 写的那本。以MINIX为例子讲解操作系统的概念。非常推荐。

2. 有了操作系统的基本概念以后，可以了解Linux的机制了。推荐《Linux内核设计与实现》

Robert Love 写的。这本书从概念上讲解了Linux有什么，他们是怎么运行的。这本书要反复认真看透。

3. 有了Linux内核的了解，还需要具体研究Linux内核源码。最经典的就是《深入理解Linux内核》Daniel P. Bovet 写的。学习这本书的时候，要对着内核代码看着学。这本书学起来相当费力了，那么多多代码要研究。不过这本书如果学明白了，恭喜你，Linux内核你已经很熟悉了。

4. 如果要开发设备驱动，可以学习《linux设备驱动程序》O'Reilly出版社的。这本作为驱动的入门是很好的资料。另外还有一本《精通 Linux 驱动程序开发》也是不错的教材，可以参考着看。学习驱动，免不了要学习一些硬件的协议和资料，研究哪个就找到相应的硬件文档，把硬件的工作原理搞明白。这些就不细说了。

5. 网络部分，学些Linux网络部分就学习《深入理解LINUX网络技术内幕》。这本书把Linux的网络部分讲的非常清晰透彻。但是通常不做这方面的工作研究，也不用研究这么深，毕竟现在国内相关职位较少。

6. 现在Linux相关的工作，多集中在一些嵌入式开发领域，arm，mips等，要学习以下这些体系架构的资料，了解CPU的设计和工作方式。ARM就看对应的芯片手册，讲的很细致。MIPS就看《see mips run》，有一二两版，两版内容有些差异，推荐都看。

7. 补充一点经验。不要认为Linux很庞大，很复杂，就觉的很难学。任何东西认真学下来都是能学会的，看你的恒心和毅力了。另外，不要走弯路，不要看市面上讲什么Linux0.11的那些书，直接学你要学的东西。就像学C语言看什么谭浩强一样，弯路走了，力气没少花，还严重影响学习效果。

你问的内核，多给你说几句应用编程，有时候经常会需要的：

1. 学习Linux应用编程，建议看《unix环境高级编程》，把里面的例子都做一遍，会对整个Linux编程有系统都认识。

2. 针对Linux，有本《Linux系统编程》，学完上一本，这本很快看一遍就懂了。主要是针对Linux

具体懂一些内容，讲的挺全了，很实用。

3. Linux网络编程，系统的学习一下《unix网络编程.卷1,套接字联网api》，基本上网络应用相关的程序就都没问题了。