## 新人如何学习嵌入式Linux

作为一个新人，怎样学习嵌入式Linux？被问过太多次，特写这篇文章来回答一下。  
在学习嵌入式Linux之前，肯定要有C语言基础。汇编基础有没有无所谓(就那么几条汇编指令，多看几遍就会，并且ARM裸机1期加强版也有讲汇编)。  
 C语言要学到什么程度呢？当然越熟越好，不熟的话也要具备基本技能。比如写一个数组排序、输入数字求和什么的。学C语言唯一的方法是多写程序多练习，编译出错没关系，自己去解决；执行出错没关系，自己去分析。以前我用VC练习C语言，经常尝试着写一些C竞赛的题目。它们是纯C、纯数学、纯逻辑的，不涉及界面这些东西，很适合煅炼编程能力。  
      
 回到主题，首先我们要明白你的目的是什么，大概来说所谓嵌入式Linux可以分为两部分：底层系统、应用开发。 如果想做应用开发，那么把C语言、数据结构、JAVA什么的学好。嵌入式应用开发和PC上的应用开发并没有什么特别要注意的。也许有人说在嵌入式上要做某些优化，是的，但是未经优化的程序和PC上的程序开发没什么差别。

另外，当你有能力去优化时，已经不用来问这个问题了。具体到某个例子，  
比如开发界面，PC上我们用VC；在嵌入式Linux里也许用Android或QT，这个时候你应该去学学QT、Android编程。但是用到的基础还是C、JAVA、c++，夯实基础后再去熟悉它们的接口。如果你的目的是学习底层系统，这是我的专长，倒是可以说多点。  
回答这个问题之前，我先解释：不少人问我的，到底是学驱动还是学应用？ 

1.1我只能说凭兴趣，并且驱动和应用并不是截然分开的

1. 我们说的驱动，其实并不局限于硬件的操作，还有操作系统的原理、进程的休眠唤醒调度等概念。 要写出一个好的应用，比较好的解决应用碰到的问题，这些知识你应该懂   
2. 做应用门槛低，特别是现在的ANDROID，纯JAVA。做应用的发展路径个人认为就是业务纯熟。  
比如在通信行业、IPTV行业、手机行业，了解行业的需求。所以，当领导的人，多是做应用出身。

3. 做驱动，其实我不想称为“做驱动”，而是想称为“做底层系统”，做好了通杀各行业。我工作几年，做过手机、IPTV、会议电视，但是这些产品对我毫无差别，因为我只做底层。他们的业务跟我没关系。 当应用出现问题，他们解决不了时，我就会从内核角度给他们出主意、提供工具。   
做底层的发展方向，个人认为是技术专家。  
4. 其实，做底层还是做应用，之间并没有一个界线，有底层经验，再去做应用，会感觉很踏实。有了业务经验，再了解一下底层，很快组成一个团队。   
     
 回到怎么学的问题上。嵌入式Linux底层系统包含哪些东西？不要急，举一个例子你就知道了。

1. 电脑一开机，那些界面是谁显示的？是BIOS，它做什么？一些自检，然后从硬盘上读入windows，并启动它。类似的，这个BIOS对应于嵌入式Linux里的bootloader。这个bootloader要去Flash上读入Linux内核，并启动它。

1. 启动windows的目的是什么？当然是上网聊天游戏。这些上网、聊天工具在哪？  
      在C盘、D盘上。所以, windows要先识别出C盘、D盘。在Linux下我们称为根文件系统。  
   3. windows能识别出C盘、D盘，那么肯定能读写硬盘才行。这涉及的东西称为驱动程序。当然不仅仅是硬盘，还有网卡、USB等等。嵌入式Linux能从Flash上读出并执行应用程序，肯定也得有Flash的驱动程序啊，当然也不仅仅是Flash。

        
嵌入式LINUX里含有bootloader, 内核, 驱动程序、根文件系统这4大块。

一、bootloader：

  它就是一个稍微复杂的裸板程序。但是要把这个裸板程序看懂写好一点都不容易。Windows下好用的工具弱化了我们的编程能力。

很多人一玩嵌入式就用ADS、KEIL。你能回答下面几个问题吗？

1. 一上电，CPU从哪里取指令执行？

   答：一般从Flash上指令。

2. 但是Flash一般是只能读不能直接写的，如果我用到全局变量，这些全局变量在哪里？  
   答：全局变量应该在内存里。

3. 那么谁把全局变量放到内存里去？

答：长期用ADS、KEIL的朋友，你能回答吗？这需要"重定位"。在ADS或KEIL里，重定位的代码是制作这些工具的公司帮你写好了。你可曾去阅读过？

4. 内存那么大，我怎么知道把"原来存在Flash上的内容"读到内存的"哪个地址去"？  
 答：这个地址用"链接脚本"决定，在ADS里是scatter文件，KEIL里也有类似的文件。但是，你去研究过吗？

5. 你说重定位是把程序从Flash复制到内存，那么这个程序可以读Flash啊？  
答：是的，要能操作Flash。当然不仅仅是这些，还有设置时钟让系统运行得更快等等。  
     
先自问自答到这里吧，bootloader这一裸板程序，其实有3部分要点：

1. 对硬件的操作

2. 对ARM体系处理器的了解

3. 程序的基本概念：重定位、栈、代码段数据段BSS段等等。

     
  对硬件的操作，需要看原理图、芯片手册，需要一定的硬件知识，不求你能设计硬件，但是至少看懂原理图; 不求能看懂模拟电路，但求能看懂数字电路。这方面的能力我是在学校里学到的，微机原理、数字电路这2本书(书名忘了)就足够了。但是我怀疑你有无耐心把这2本书看完。我不知道现在有没有更快捷的书。想速成的话，先放掉这块吧，不懂就问GOOGLE、发贴。另外，肯定要读芯片手册，别去找中文的，就看英文的。刚开始非常痛苦，一旦熟悉，读任何芯片手册都很容易。对ARM体系处理器的了解, 看杜春蕾的<ARM体系架构与编程>，里面讲了汇编指令，有异常模式、MMU等。也就这3块内容需要你了解。

程序的基本概念，王道当然是去看编译原理了。可惜这类书绝对是天书级别的。劝你若非超级天才还是别去看了。就看我写的《嵌入式Linux应用开发完全手册》和第1期视频吧，别担心，不用花钱。照着视频把硬件相关的实验做了，这些概念就清楚了。还没有  
发现第2套讲这些概念的书或视频，允许我盲目吹嘘一回。

对于bootloader，我学习时是先看了《ARM体系架构与编程》，然后自己写程序把各个硬件的实验都做了一遍，比如GPIO、时钟、SDRAM、UART、NAND。把它们都弄清楚了，组合在一起就很容易看懂u-boot了。

总结一下，看懂硬件原理图、看芯片手册，这需要你自己去找资料。剩下的，就按《嵌入式Linux应用开发完全手册》和第1期视频的章节目录学习。

二、内核：

想速成的人，先跨过内核的学习，直接学习怎么写驱动。

想成为高手，内核必须深刻了解。注意，我说的是了解，没奢望去写出一个内核。

要对里面的调度机制、内存管理机制、文件管理机制等等有所了解。

推荐两本书：

1. 通读《linux内核完全注释》，请看薄的那本(浮燥的社会讲求速度, 呵),

2. 选读《Linux内核源代码情景分析》, 了解哪一块就读哪一节

三、驱动：

驱动包含两部分：硬件本身的操作、驱动程序的框架。

又是硬件，还是要看得懂原理图、读得懂芯片手册，多练吧。

说到驱动框架，有一些书介绍一下。LDD3即《Linux设备驱动》，老外写的那本，里面介绍了不少概念，值得一读。但是它的作用也就限于介绍概念，基本上是入门之前用它来熟悉一些概念，入门后就扔掉了。

驱动方面比较全的书，应该是宋宝华的《linux设备驱动开发详解》，老实说只看过目录，不少人说好，这里推荐一下。要想深入了解某一块，《Linux内核源代码情景分析》绝对是超5星级推荐。别指望把它读完。全书上下两册1800多页呢。某一块不清楚，翻一下。任何一部分，它都都可以讲上2、3百页，非常详细。并且以某个目标带你分析内核源码。它以linux 2.4为例，但是原理相通，同样适用于其它版本的linux。

四、根文件系统：

大家有没有想过这2个问题：

1. 对于Linux做出来的产品，有些用作监控、有些做手机、有些做平板。那么内核启动后，挂载根文件系统后，应该启动哪一个应用程序呢？

答：内核不知道也不管应该启动哪一个用户程序。它只启动init这一个应用程序，它对应/sbin/init。显然，这个应用程序就要读取配置文件，根据配置文件去启动用户程序(监控、手册界面、平板界面等等)，这个问题提示我们，文件系统的内容是有一些约定的，比如要有/sbin/init，要有配置文件。

2. 你写的hello,world程序，有没有想过里面用到的printf是谁实现的？  
 答：这个函数不是你实现的，是库函数实现的。它运行时得找到库。  
 这个问题提示我们，文件系统里还要有库。

         
 简单的自问自答到这里，要想深入了解，可以看一下busybox的init.c，可以知道init进程做了哪些事情。当然也可以看《嵌入式Linux应用开发完全手册》构建根文件系统那章。

说一下我的学习经历。

1. 在学校读的是物理电子专业，其实课程里没有教怎么设计电路，只是教了些电子电路方面的知识。PCB的设计是在实验室里自学的，只设计过2层板，现在忘得差不多了。但是保留了看原理图、看芯片手册的能力。

2. 选修了软件学位，对软件设计挺感兴趣，但是也只是学了C语言、数据库而已。凭着兴趣做了不少竞赛题。没能力去参加竞赛，但是把C练得很扎实。

3. 在实验室、在第1家公司，就是设计些简单的PCI卡，写一下windows的驱动程序  
4. 在第2家公司，用51单片机做车载电话，开始走上纯软件的道路。

5. 开始感到单片机的不足，辞职半年闭门学Linux，从red hat怎么操作开始。步骤就是先看《ARM体系架构与编程》，再写裸板程序操作硬件，接着分析u-boot。同时看《linux内核完全注释》，对LINUX框架有所了解。在写裸板时建议各位加强对中断的理解，内核就是用中断来完成各种功能的。

6. 分析完u-boot，开始进行简单的驱动编程，这时候能力还很弱。

开始去中兴上班2年，编写各类驱动、解决各类问题(驱动问题、帮助定位应用问题)，能力得到煅炼。

      
 总结一下要看的书：

1. 硬件方面: 微机原理、数字电路，了解即可，不需要深入

。  
2. Linux方面：

《ARM体系架构与编程》杜春雷

《嵌入式Linux应用开发完全手册》韦东山

老外写的《Linux设备驱动》ldd3

《linux设备驱动开发详解》宋宝华

《linux内核完全注释》赵炯

《Linux内核源代码情景分析》毛德操