跟涛哥一起学嵌入式 15：你为什么看不懂Linux内核驱动源码？

文档说明	作者	日期
来自微信公众号：宅学部落(armLinuxfun)	wit	2019.11.26
嵌入式视频教程淘宝店：https://wanglitao.taobao.com/		
联系微信：brotau(宅学部落)		

学习嵌入式Linux驱动开发，最核心的技能就是能够编写Linux内核驱动、深入理解Linux内核。而做到这一步的基础，就是你要看得懂Linux内核源码，了解其基本的框架和具体实现，了解其内核API的使用方法，然后才能根据自己的需求写出高质量的内核驱动程序。

说易行难，很多新人、甚至工作1-2年的开发者刚接触Linux内核时，别说写了，看内核代码可能都是一脸懵逼：明明是C语言，但是就是看不懂是什么意思，除了根据函数名、函数参数、函数的返回值以及注释，了解整个函数的基本功能外，一旦分析其细节，你会发现，寸步难行，每一行代码似乎都深不可测，仿佛蕴含着极大的陷阱和能量，于是你翻书、百度、Google，一步一步地向前推进。

这是很多初学者常用的学习方法，包括我在内，刚接触Linux内核驱动时，看到很多似曾相识，但仔细推敲起来又很陌生的内核代码，内心是崩溃的、内心是虚的：乖乖，这是啥玩意儿，为什么看不懂，难道我智商有问题？就我一个人看不懂吗？甚至工作后我曾请教过很多工作经验的同事，想请教一些关于阅读内核的经验和方法，你会发现，他们很多其实对内核也没深入研究，除了自己负责的模块比较熟悉外(寄存器配置，数据流程)，对于其它的模块也很少有时间和精力去关注，甚至关于本模块的框架也很少关注，当然，这大部分是精力和时间的关系，工作量的需求要求你快速通过各种API完成任务。甚至有同事会说，看Linux内核就像隔雾看花，朦胧一点比较好，不能细看，越细看越不懂。

但是，要想编写高质量的程序，对Linux内核、模块的理解肯定要深入的，否则，你就永远停留在外围，缝缝补补。根据我的各种学习经历和经验总结，Linux内核并不是坚不可摧、攻不可破，掌握了正确的学习方法和知识基础，我们也可以在内核的代码里遨游，领略Linux内核中C语言的各种奇妙应用和强大技巧，对Linux内核各种复杂的框架、子系统也可以指点江山，胸有成竹。

那学习Linux驱动、分析Linux内核源码之前，到底需要哪些知识储备和技能呢？

1）C语言基础+数据结构

Linux内核，好不夸张地说，就是由各种结构体、函数指针、链表、队列堆彻而成的。所以在进军Linux内核之前，你的C语言基础一定要打牢固：什么函数指针、指针函数、数组指针、指针数组、以及各种指针作为函数参数、返回值等等都要搞清楚，因为Linux内核中大量使用这些。这些都是基础，现在犯迷糊，看内核更是晕。

除此之外，数据结构也是要掌握的，链表、队列在Linux内核中大量使用，所以必须要掌握。像其它的一些非线性数据结构：比如树、二叉树、红黑树等，对于做底层驱动的开发者来说，接触得很少，可以先不学，用到的时候再补也不迟。

2）C语言的语法扩展

在阅读Linux内核代码的过程中，你有没有感觉到，有些代码，看起来“怪怪的”，跟一般的C语言不太一样？看起来是C语言，仔细一分析，发现又看不懂了。

这些你看起来“怪怪的C语言代码”，其实都是GCC编译器对标准C语言的扩展语法：比如语句表达式、局部标签、**attribute**属性声明、可变参数宏等。这些GCC扩展的语法，在Linux内核、驱动源码中，广泛使用，尤其是涉及到底层启动、编译链接的一些设置。如果你不掌握这些扩展的C语言语法的使用，在阅读Linux内核源码、或驱动的过程中，可能就会遇到很多障碍，跟着跟着就没路了，断篇了，对我们理解代码造成各种干扰。

所以在打算阅读Linux源码之前，建议先学习下GNU C对标准C的常用扩展语法

学完了这个，扫除了阅读Linux源码的语法障碍，接下来可以选择一个自己感兴趣的小模块：先把这个模块留给用户的API玩熟，学会编程，再慢慢研究其内核内部的实现。从底层到上层，打通任督二脉，再去分析内核中，其它复杂的系统，也就触类旁通，比较容易上手了。

3）Linux内核中的面向对象思想

有了上面的基础，我们分析一个小的Linux内核模块，是没有问题的。当遇到一个大的复杂子系统，比如说USB子系统、内存管理、MTD、文件系统等，结构体里面嵌套多层结构体，各种device、bus、driver、各种层，是不是有点绕晕了？有种盲人摸象、在森林里迷路的感觉，把握不了“全局”。

这时候，我们就不能使用C语言的面向过程思维了，Linux内核的设计其实大量使用了面向对象思想、设计模式。因此，我们要学会用面向对象的思维去分析Linux内核，分析各个模块的复用，这样就很方便的在脑海中搭建出系统的框架和层次了。然后再使用面向过程思维去分析具体的功能实现、具体细节，多花点时间和精力，相信你会有不一样的收获的。

4）《C语言嵌入式Linux高级编程》视频教程简介

《C语言嵌入式linux高级编程》视频教程，主要针对很多嵌入式学员来自不同专业、知识架构和体系欠缺这一背景，着重讲解嵌入式开发中需要的理论知识和必备技能：

- 计算机系统结构与原理
- ARM结构与汇编语言编程
- 程序的编译、链接和运行原理
- 堆栈内存管理、堆栈溢出攻击原理、内存泄露

- Linux内核中的GNU C扩展语法
- C语言指针深入详解
- 嵌入式中常用的数据结构
- Linux内核中的面向对象思想
- C语言的模块化编程
- CPU和操作系统入门

这是一条全新的嵌入式C语言进阶路线，**独家录制**，**网上独此一家！一线芯片驱动开发工程师耗时一年精心打造**！无论是专业角度、实用性、还是深度广度上，都作了很大的改进和提升。以含金量和专业深度提高你的C语言理论基础和编程技能。

无论您是嵌入式初学者，还是工作1~3年的职场新兵，学习完本课程，可以弥补嵌入式开发所需要的专业壁垒和核心理论短板，为成为一个嵌入式高手奠定深厚的内功基础。

教程淘宝店：https://item.taobao.com/item.htm?spm=a2oq0.12575281.0.0.25911debvB6WAY&ft=t&id=577829845886

专注嵌入式、Linux精品教程：https://wanglitao.taobao.com/
嵌入式技术教程博客：http://zhaixue.cc/
联系 QQ：3284757626
嵌入式技术交流QQ群：475504428
微信公众号：宅学部落(armlinuxfun)