

# 直到我干了底层开发，才知道不写业务代码有多爽

原创 Vincenterr 嵌入式Linux充电站 2023年07月19日 08:15 广东

点击上方“[嵌入式Linux充电站](#)”，选择“[置顶/星标公众号](#)”

福利干货，第一时间送达

写业务代码，可能每天大部分时间调别人的接口，接口不合适封装个函数，需求也不断变动。

**总归实现功能最重要，用户体验最重要，别人提需求，你只管实现。**

你就像无情的机器，写重复的业务代码，实现相似的功能，感觉没长进，眼界也没提升.....

**但驱动开发截然不同.....**

## — 01 —

### 行不行，你说了算

业务代码，最多的就是需求变动。

例如，这个模块再加个功能，这个代码再封装一下，你这样写不行要这样写.....

**但是，对于一个驱动来说，它能用就是能用，不能用就是不能用，没有太多的需求可言。**

例如一个网卡驱动，它能正常连接网络，能正常传输文件就是它的功能，代码怎么写是根据芯片手册来的，没人会对你提什么需求。

所以，没有需求的烦恼。只要能用，你的任务就完成了，不会有什么需求过来，最多就是改一下bug。

## — 02 —

### 除了你，一般人不敢动你的代码

#### 为什么不敢动？

驱动代码不像应用层写的业务逻辑，业务代码可能更多的是一些编程技巧、代码优化上的东西，例如接口封装、参数判断、特殊情况处理等一些偏软件设计上的东西，如果看到某段业务逻辑写的不好，有些经验的人都是比较容易去改你的代码的。

**但写驱动代码有一个最大特点，就是先把芯片手册看懂。**

例如驱动里分配了一个64字节对齐的内存、调度前又设置了一遍寄存器，为什么要这么做？你是不知道的。

**如果别人觉得你的驱动有问题，想改你的代码，那他应该先把芯片手册看懂。**

没看过芯片手册之前，这个代码他没法动，这个名词、这个比特位是什么意思，代码为什么这么写，只有你才知道。

即便是驱动真有问题，也一定是你自己改，因为一般人不会闲着没事去看不属于他工作范围的文档，况且把文档看懂都要好几天。

即便是一处很小的错误，通常也是你自己改，因为别人不知道改了这个地方，会不会影响其他地方，只有写这个驱动的人才知道。

所以只要驱动能正常用，就不太会有人去动它了。

**我告诉你我有什么功能，**

**而不是你来告诉我你要什么功能**

如果是业务代码，通常都是需求牵着你走，一切为了用户体验。

但驱动代码相反，**我只会告诉你，我有什么功能，而不是你来告诉我，你要什么功能。**

一个驱动，它的功能就只有这些，只能少不能多，我可以告诉你这个驱动有这些功能，你想要什么样的接口，我可以提供ioctl接口或者其他接口给你调用。

如果你说你要传一个参数，把xxx返回给你，我也有可能跟你说，做不到。如果你说你要同时传一些参数，把xxx同时返回，我也只会告诉你做不到。

你问为什么？因为这个硬件本身就做不到这样的效果。**如果你不信，那你去把芯片手册看一遍吧，看懂再来问我。**

end

### 往期推荐

你解决bug的能力，暴露了你的水平

入职Linux驱动工程师后，我才知道的真相.....

很底层的性能优化：让CPU更快地执行你的代码

机遇：我是如何走向Linux驱动的...

当我用几道题考了一遍做Linux驱动的同事.....



## 嵌入式Linux充电站

作者Vincent，分享一些嵌入式Linux、内核、RISC-V等知识。学习、沉淀、分享，才能有...

100篇原创内容

公众号

程序人生 17

程序人生 · 目录

上一篇

新手入职Linux驱动岗后，必听的5个忠告

下一篇

成为驱动工程师后的第一课：