

基于Linux操作系统核心的汉字显示（5）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E5_9F_BA_E4_BA_8E_Linu_c103_144358.htm 也许有读者会问，只需要为每个字符添加一位信息来标志是否是汉字就足够了，为什么还要开辟与原缓冲区大小相同的双倍缓冲区，这是不是太浪费呢？我们先放下这个问题，稍后再作回答。其实，如果再添加一位来标志当前字符是汉字的左半边还是右半边的话，就会省去扫描屏幕上当前整行字符串的工作，这样一来，编程会更简单，但是有读者会问，即使是这样，使用8位总够用了吧？为什么还要使用16位呢？我们的做法是：用低8位来存放汉字另外一半的内码，用高8位中的2位来存放上面所讲的辅助信息，高8位的剩余6位可以用来存放汉字或其他编码方式（如BIG5或日文、韩文）的信息，从而使我们可以实现同屏显示多种双字节语言的字符而不会相互干扰。另外，在编程时，双倍缓冲也比较容易计算。这样我们就回答了如上的两个问题。迄今为止，我们有了一套彻底解决汉字和制表符相互干扰，半个汉字的刷新、重绘等问题的方案。剩下的就是具体编程来实现的问题了。但是，由于Framebuffer的驱动程序很多，修改每一个驱动程序的xxxx_putc()函数和xxxx_putcs()函数会是一项不小的工作，而且，改动驱动程序后，每种驱动程序的测试也是很麻烦的，尤其是对于有硬件加速的显卡，修改和测试会更不容易。那么，是否存在一种不需要修改显卡驱动程序的方法呢？经过一番努力，我们发现，可以调用xxxx_putcs()或xxxx_putc()函数输出汉字之前，修改VGA字库指针使其指向所需显示的汉字在汉字字库中

的位置，即把一个汉字当成两个VGA ASCII字符输出。也就是说，在内核中存在两个字库，一个是原有的VGA字符字库，另一个是汉字字库，当我们需要输出汉字的时候，就把VGA字库的指针指向汉字字库的相应位置，汉字输出完之后，再把该指针指向VGA字库的原有位置。这样一来，我们就只需要修改fbcon.c和console.c，其中console.c负责维护双倍缓冲区，把每一个字符的信息存入附加的缓冲区中；而fbcon.c负责利用双倍缓冲区中的附加的信息，调理VGA字库的指针，调用底层的显示驱动程序。这里还有几个需要注意的地方：1、由于屏幕重绘等原因，调用底层xxxx_putc()和xxxx_putc()的地方有多处，我们做了两个函数分别替换这两处调用，完成替换字库、调用xxxx_putcs()或xxxx_putc()、恢复字库等功能。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com