

基于Linux操作系统核心的汉字显示（3）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\\_144362.htm](https://www.100test.com/kao_ti2020/144/2021_2022__E5_9F_BA_E4_BA_8E_Linu_c103_144362.htm) 显示中文 比如说我们试输出一句中文：`putcs(你好)`（“你好”的内码为0xc4.0xe3.0ba.0xc3）。这时候会怎么样呢？有一点可以肯定，“你好”肯定不会出现在屏幕上，原因是：1、核心中没有汉字字库，中文显示就是无米之炊了。2、在负责字符显示的void `fbcon_cfb8_putcs()`函数中，原有操作如下：对于每个要显示的字符，依次从虚拟终端缓冲区中以WORD为单位读取（低位字节是ASCII码，高8位是字符的属性）。由于汉字是双字节编码方式，所以这种操作是不可能显示出汉字的，只能显示xxxx\_putcs()输出的是一个一个的VGA字符。因此，要解决的问题：确保在调用`do_con_write()`时进行uni\_pc转换不会改变原有编码，一个很直接的实现方式就是加载一个我们自己定制的Unicode映射表，`loadunimap direct.uni`，或者进接把`direct.uni`设置为核心的缺省映射表。针对以上问题，我们要做的第一个尝试方案如下：首先需要在核心中加载汉字字库，然后修改`fbcon_cfb8_putcs()`函数，在`fbcon_cfb8_putcs()`中一次读两个WORD，检查这两个WORD的低位字节是否能拼成一个汉字，如果发现能拼成一个汉字，就算出这个汉字在汉字字库的的偏移，然后把它当成个16×16的VGA字符来显示。试验的结果表明：1、能够输出汉字，但仍有许多不理想的地方，比如说，输出以半个汉字开始的一串汉字，则这半个汉字后面的汉字都会是乱码，这是“半个汉字”的问题。2、光标移动会破坏汉字的

显示，表现为，光标移动过的汉字会变成乱码，这是因为光标的更新是通过xxxx\_putc()函数来完成的。xxxx\_putc()函数与xxxx\_putcs()函数实现的功能类似，但是xxxx\_()函数只刷新一个字符而不是一个字符串，因而xxxx\_putc()的输入参数是一个整数，而不是一个字符串的地址，xxxx\_putc()函数的声明如下：`void fbcon_cfb8_putc(struct vc_data *conp, struct display *p, int c, int YY, int xx)`下一个尝试方案就是同时修改xxxx\_putc()函数和xxxx\_putc()函数为了解决半个汉字的问题，每一次输出之前，都从屏幕当前行的起始位置开始打措，以确定要输出的字符是否落在半个汉字的位置上，如果是在半个汉字的位置上，如果是在半个汉字的位置，则进行相应的调整，即从向前移动一个字节的位置开始输出。 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](http://www.100test.com)