

基于Linux系统核心的汉字显示的尝试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E5_9F_BA_E4_BA_8E_Linu_c103_145158.htm 在阐述基于Linux核心的汉字显示的技术细节之前，有必要介绍一下原有linux的工作机制。这里主要涉及到两部分的知识，就是Linux下终端和帧缓冲的实现。控制台（console）通常我们在linux下看到的控制台(console)是由几个设备完成的。分别是/dev/ttyN（其中tty0就是/dev/console，tty1，tty2就是不同的虚拟终端（virtual console））。通常使用热键alt Fn来在这些虚拟终端之间进行切换。所有的这些tty设备都是由linux/drivers/char/console.c和vt.c对应。其中console.c负责绘制屏幕上的字符，vt.c负责管理不同的虚拟终端，并且负责提供console.c需要绘制的内容。Vt.c把不同虚拟终端下需要交给console.c绘制的内容放到不同的缓存中去。Vt.c管理着这样一个缓冲区的数组，并且负责在其间切换，以指定哪一个缓冲区是被激活的。你所看到的虚拟终端就对应着被激活的缓冲区。Console.c同时也负责接收终端的输入，然后把接收到的输入放到缓冲区。帧缓冲

（framebuffer）Framebuffer是把显存抽象后的一种设备，可以通过这个设备的读写直接对显存进行操作。这种操作是抽象的，统一的。用户不必关心物理显存的位置、换页机制等等具体细节。这些都是由Framebuffer设备驱动来完成的。

Framebuffer对应的源文件在linux/drivers/video/目录下。总的抽象设备文件为fbcon.c，在这个目录下还有与各种显卡驱动相关的源文件。在使用帧缓冲时，Linux是将显卡置于图形模式下的。试验 我们以一个简单的例子来说明字符显示的过程。

我们假设是在虚拟终端1 (/dev/tty1) 下运行一个如下的简单程序。 `main () { puts("hello, world.\n"). }`puts函数向缺省输出文件(/dev/tty1)发出写的系统调用write(2)。系统调用到linux核心里面对应的核心函数是console.c中的con_write() , con_write() 最终会调用do_con_write()。在do_con_write()中负责把"hello, world.\n"这个字符串放到tty1对应的缓冲区中去。

do_con_write()还负责处理控制字符和光标的位置。让我们来看一下do_con_write()这个函数的声明。 `static int do_con_write(struct tty_struct * tty, int from_user, const unsigned char *buf, int count)` 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