

加入QQ群

昵称： 夏大王

园龄： 10年10个月

粉丝： 45

关注： 18

+加关注

< 2011年3月 >						
日	一	二	三	四	五	六
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找我

谷歌搜索

- 常用链接
- 我的随笔

我的评论

我的参与

最新评论

我的标签

- 我的标签
- android(8)

java(3)

Linux(3)

智能指针(2)

socketcan(2)

相册(1)

系统相机(1)

android初级篇(1)

Linux的帧缓冲设备(Framebuffer)简介

帧缓冲 (framebuffer) 是Linux为显示设备提供的一个接口，把显存抽象后的一种设备，他允许上层应用程序在图形模式下直接对显示缓冲区进行读写操作。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。

帧缓冲驱动的应用广泛，在linux的桌面系统中，Xwindow服务器就是利用帧缓冲进行窗口的绘制。尤其是通过帧缓冲可显示汉字点阵，成为Linux汉化的唯一可行方案。

Linux FrameBuffer 本质上只是提供了对图形设备的硬件抽象，在开发者看来，FrameBuffer 是一块显示缓存，往显示缓存中写入特定格式的数据就意味着向屏幕输出内容。所以说FrameBuffer就是一块白板。**例如对于初始化为16 位色的FrameBuffer 来说，FrameBuffer中的两个字节代表屏幕上一个点，从上到下，从左至右，屏幕位置与内存地址是顺序的线性关系。**

帧缓存可以在系统存储器(内存)的任意位置，视频控制器通过访问帧缓存来刷新屏幕。 帧缓存也叫刷新缓存 Frame buffer 或 refresh buffer, 这里的帧(frame)是指整个屏幕范围。

帧缓存有个地址，是在内存里。我们通过不停的向frame buffer中写入数据， 显示控制器就自动的从frame buffer中取数据并显示出来。全部的图形都共享内存中同一个帧缓存。

CPU指定显示控制器工作，则显示控制器根据CPU的控制到指定的地方去取数据和 指令， 目前的数据一般是从显存里取， 如果显存里存不下，则从内存里取， 内存也放不下，则从硬盘里取，当然也不是内存放不下，而是为了节省内存的话，可以放在硬盘里，然后通过 指令控制显示控制器去取。帧缓存 Frame Buffer，里面存储的东西是一帧一帧的， 显卡会不停的刷新Frame Buffer，这每一帧如果不捕获的话， 则会被丢弃，也就是说是实时的。这每一帧不管是保存在内存还是显存里， 都是一个显性的信息，这每一帧假设是800x600的分辨率， 则保存的是800x600个像素点，和颜色值。

显示器可以显示无限种颜色，目前普通电脑的显卡可以显示32位真彩、24位真彩、16位增强色、256色。除256色外，大家可以根据自己的需要在显卡的允许范围之内随意选择。很多用户有一种错误概念，认为256色是最高级的选项，而实际上正好相反。256色是最低级的选项，它已不能满足彩色图像的显示需要。16位不是16种颜色，而是2的16次平方(256×256)种颜色，但256色就是256(2的8次平方)种颜色。所以16位色要比256色丰富得多。

帧缓冲设备对应的设备文件为/dev/fb*，如果系统有多个显示卡，Linux下还可支持多个帧缓冲设备，最多可达32 个，分别为/dev/fb0到/dev/fb31，而/dev/fb则为当前缺省的帧缓冲设备，通常指向/dev/fb0。当然在嵌入式系统中支持一个显示设备就够了。帧缓冲设备为标准字符设备，主设备号为29，次设备号则从0到31。分别对应/dev/fb0- /dev/fb31。通过/dev/fb，应用程序的操作主要有这几种：

- 1．读/写 (read/write) /dev/fb：相当于读/写屏幕缓冲区。例如用 cp /dev/fb0 tmp命令可将当前屏幕的内容拷贝到一个文件中，而命令cp tmp > /dev/fb0 则将图形文件tmp显示在屏幕上。
- 2．映射 (map) 操作：由于Linux工作在保护模式，每个应用程序都有自己的虚拟地址空间，在应用程序中是不能直接访问物理缓冲区地址的。为此，Linux在文件操作 file_operations结构中提供了mmap函数，可将文件的内容映射到用户空间。对于帧缓冲设备，则可通过映射操作，可将屏幕缓冲区的物理地址映射到用户空间的一段虚拟地址中，之后用户就可以通过读写这段虚拟地址访问屏幕缓冲区，在屏幕上绘图了。实际上，使用帧缓冲设备的应用程序都是通过映射操作来显示图形的。由于映射操作都是由内核来完成，下面我们将看到，帧缓冲驱动留给开发人员的工作并不多。
- 3．I/O控制：对于帧缓冲设备，对设备文件的ioctl操作可读取/设置显示设备及屏幕的参数，如分辨率，显示颜色数，屏幕大小等等。ioctl的操作是由底层的驱动程序来完成的。

在应用程序中，操作/dev/fb的一般步骤如下：

- 1．打开/dev/fb设备文件。
- 2．用ioctl操作取得当前显示屏幕的参数，如屏幕分辨率，每个像素点的比特数。根据屏幕参数可计算屏幕缓冲区的大小。
- 3．将屏幕缓冲区映射到用户空间。
- 4．映射后就可以直接读写屏幕缓冲区，进行绘图和图片显示了。

典型程序段如下：

```
#include <linux/fb.h>
int main()
{
    int fbfd = 0;
    struct fb_var_screeninfo vinfo;
    struct fb_fix_screeninfo finfo;
    long int screensize = 0;
    /*打开设备文件*/
    fbfd = open("/dev/fb0", O_RDWR);
    /*取得屏幕相关参数*/
    ioctl(fbfd, FBIOGET_FSCREENINFO, &finfo);
    ioctl(fbfd, FBIOGET_VSCREENINFO, &vinfo);
    /*计算屏幕缓冲区大小*/
    screensize = vinfo.xres * vinfo.yres * vinfo.bits_per_pixel / 8;
    /*映射屏幕缓冲区到用户地址空间*/
    fbp=(char*)mmap(0,screensize,PROT_READ|PROT_WRITE,MAP_SHARED, fbfd, 0);
    /*下面可通过fbp指针读写缓冲区*/
}
```

handle(1)
一分钟后自动重启(1)
更多

随笔分类

Android(41)
android 应用编程(20)
android安全(6)
C++(16)
gcc编译器(5)
Hardware : PCB知识(2)
linux GUI(2)
Linux 常用命令工具(19)
Linux 应用编程(30)
linux内核的配置 编译(6)
Linux内核基础知识(9)
Linux驱动(17)
Linux系统的配置相关(16)
linux下程序的编译及调试(16)
Linux下开发环境的搭建(7)
更多


随笔档案

2018年2月(1)
2015年11月(4)
2015年10月(24)
2015年9月(4)
2015年8月(2)
2015年1月(5)
2014年12月(1)
2014年9月(3)
2014年8月(3)
2014年7月(13)

转自：http://blogold.chinaunix.net/u1/51097/showart_498952.html

分类: UI部分

好文要顶 关注我 收藏该文 微博 微信



夏大王

关注 - 18

粉丝 - 45

+加关注


« 上一篇： 职场工程师们需注意的饮酒及点菜礼仪
» 下一篇： 深入C++的new

0 0

推荐 反对

posted @ 2011-03-02 23:07 夏大王 阅读(1464) 评论(0) 编辑 收藏 举报

刷新评论 刷新页面 返回顶部

 登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

编辑推荐：

- [NET 与树莓派] 用 MPD 制作数字音乐播放器
- 3D 穿梭效果？使用 CSS 轻松搞定
- Asp.net core 配置信息读取的源码分析梳理
- [WPF] 玩玩彩虹文字及动画
- 记一次 .NET 某风控管理系统 内存泄漏分析

最新新闻：

- 芯片危机启示录：一场重塑汽车商业模式的突变（ 2021-11-12 14:42 ）
- 雷蛇风行 RGB 面罩评测：戴上它，我就是最「亮」的仔（ 2021-11-12 14:24 ）
- 双十一死于2021（ 2021-11-12 14:10 ）
- 除了锤子，张瑞敏还留下什么？（ 2021-11-12 13:58 ）
- iOS 15 最具人文关怀的功能「数字遗产」，来了（ 2021-11-12 13:46 ）

» 更多新闻...

2014年6月(3)
2014年5月(2)
2014年3月(4)
2013年7月(3)
2013年4月(1)
更多

技术牛人blog
北漂IT民工的博客
外刊IT评论
陶显芳

阅读排行榜
1. pthread线程的终止退出 线程的大量创建(15187)
2. 常用的 Nand Flash 指令如下 : (9086)
3. ANSI编码对照表(7750)
4. 交叉编译时候如何配置连接库的搜索路径(5985)
5. android 学习笔记 (八) android底层学习8.1 学习makefile 走读build/core/main.mk(5528)

评论排行榜
1. android 学习笔记 (三) Android学习路线之3.3程序员技术练级攻略(4)
2. 学会这十步，工作不再累！你信不信，由你，反正我是信了！(2)
3. android模拟器使用SD卡(1)
4. 计算机界的传奇人物：高德纳(1)
5. 进程vs线程(1)

推荐排行榜
1. android 学习笔记 (三) Android学习路线之3.3程序员技术练级攻略(4)
2. 学会这十步，工作不再累！你信不信，由你，反正我是信了！(4)
3. 交叉编译时候如何配置连接库的搜索路径(2)

4. android控件的属性(1)
5. android 学习笔记 (八) building system8.4 android源码目录结构 (下) (1)

最新评论
1. Re:如何看懂man page ?
妈的，好后悔没有早点看到这篇文章，不过也许，现在看到也不晚
--RendezvousAuParadis
2. Re:linux驱动学习笔记 (2.3) scull模块 init_Mutex被废除
真心好文章！
--kangear
3. Re:学会这十步，工作不再累！你信不信，由你，反正我是信了！
是啊！
--maanshancss
4. Re:学会这十步，工作不再累！你信不信，由你，反正我是信了！
写的很有道理
--dengchendeng
5. Re:android 学习笔记 (三) Android 学习路线之3.3程序员技术练级攻略
不错，给力
--死神x