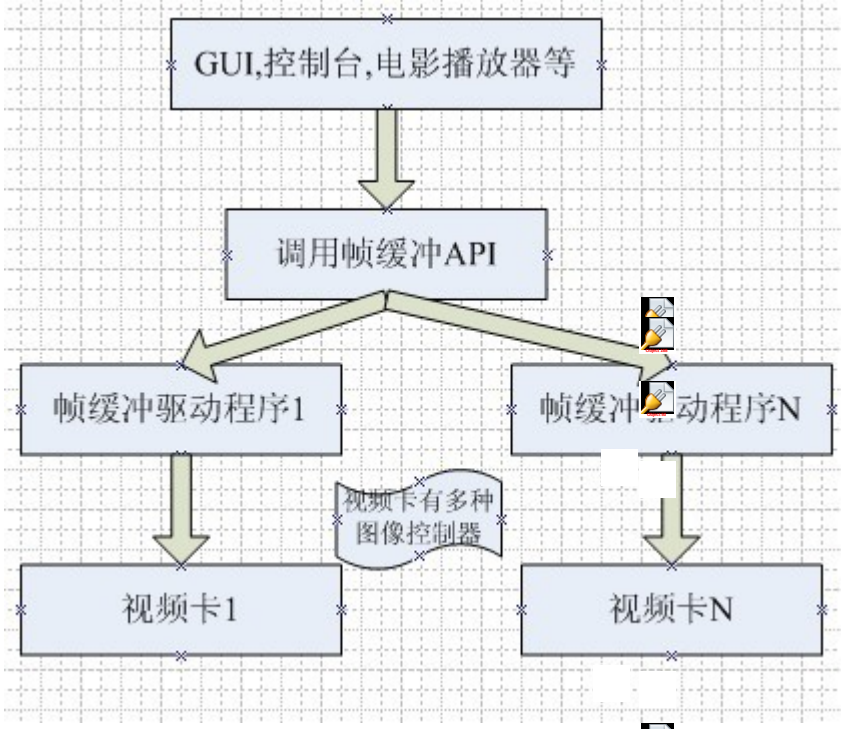
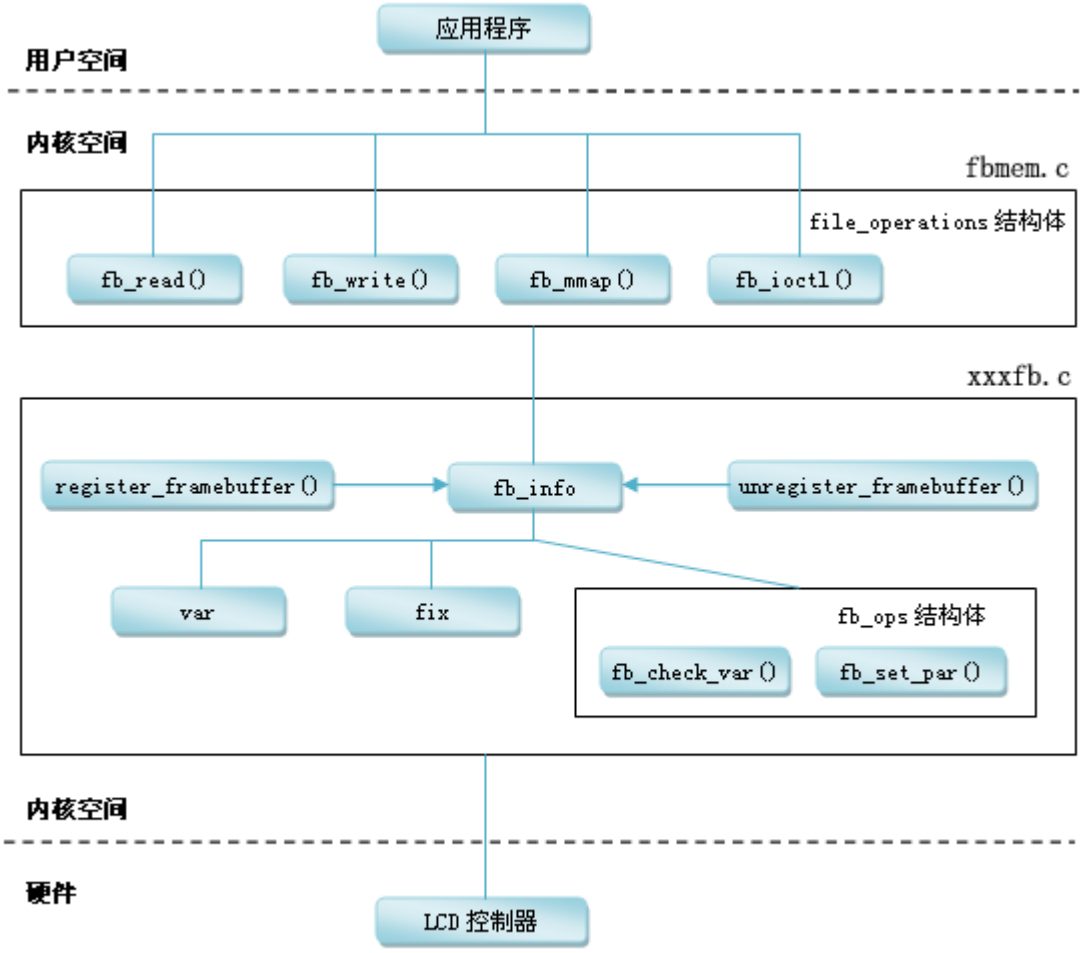
努力成为 linux kernel hacker 的人李万鹏原创作品，为梦而战。转载请标明出处

http://blog.csdn.net/woshixingaaa/archive/2011/05/29/6452688.aspx

帧缓冲(frame buffer)是 Linux 视频系统的核心概念，因此先了解一下他的功能。因为视频适配器可能基于不同的硬件体系架构，较高内核层和应用程序的实现可能会因视频卡的不同而不同，这会导致在使用不同视频卡的时需要采用不同的方案。随之而来的低可移植性和冗余的代码需要大量的投入和维护开销。帧缓冲的概念解决了这个问题，它进行了一般化的抽象并规定编程接口，从而开发人员可以以与平台无关的方式编写应用层和较高内核层程序。因此，内核的帧缓冲接口允许应用程序与底层图形硬件的变化无关，如果应用和显示器驱动程序遵循帧缓冲接口，应用程序不用改变就可以在不同类型的视频硬件上运行。



s3c2410 在 Linux 系统中的 framebuffer 驱动框架：



帧缓冲设备驱动程序结构图

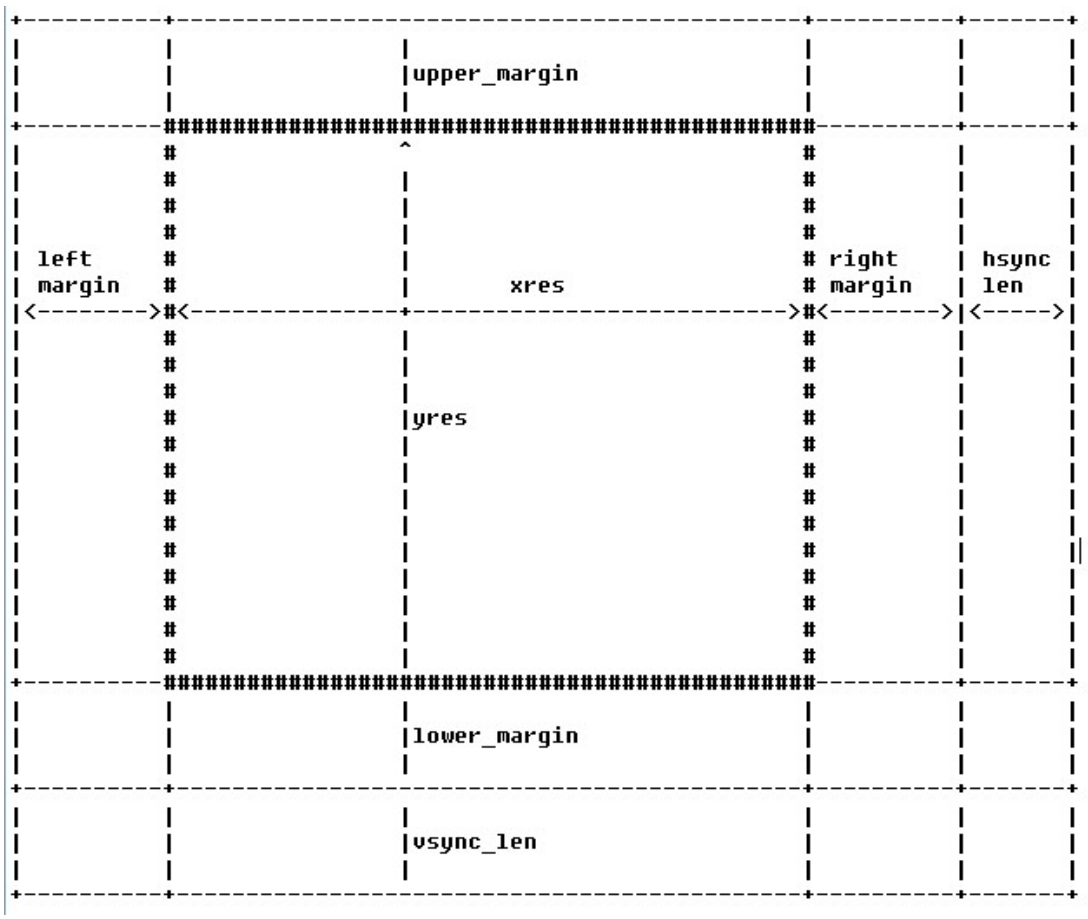
再来看 framebuffer 中用到的各种数据结构：

```
1. struct fb_fix_screeninfo {
2.     char id[16];            /*字符串形式的标识符*/
3.     unsigned long smem_start;   /*fb 缓冲内存的开始位置(物理地址)*/
4.     __u32 smem_len;         /*fb 缓冲的长度*/
5.     __u32 type;         /*FB_TYPE_* */
6.     __u32 type_aux;         /*Interleave*/
7.     __u32 visual;           /*FB_VISUAL_* */
8.     __u16 xpanstep;         /*如果没有硬件 panning，赋 0*/
9.     __u16 ypanstep;
10.    __u16 ywrapstep;
11.    __u32 line_length;      /*一行的字节数*/
12.    unsigned long mmio_start;   /*内存映射 I/O 的开始位置*/
13.    __u32 mmio_len;         /*内存映射 I/O 的长度*/
14.    __u32 accel;
15.    __u16 reserved[3];      /*保留*/
16.};
```

```
1. struct fb_var_screeninfo {
```

```
    /*可见解析度*/
    __u32 xres;
    __u32 yres;
    /*虚拟解析度*/
    __u32 xres_virtual;
    __u32 yres_virtual;
    /*虚拟到可见之间的偏移*/
    __u32 xoffset;
    __u32 yoffset;
    __u32 bits_per_pixel;       /*每像素的位数,BPP*/
    __u32 grayscale;        /*非 0 时指灰度*/
    /*fb 缓存的 R/G/B 位域*/
    struct fb_bitfield red;
    struct fb_bitfield green;
    struct fb_bitfield blue;
    struct fb_bitfield transp;  /*透明度*/
    __u32 nonstd;           /*!=0 非标准像素格式*/
    __u32 activate;
    __u32 height;           /*高度*/
    __u32 width;            /*宽度*/
    __u32 accel_flags;
    /*除 pixclock 本身外，其他都以像素时钟为单位*/
    __u32 pixclock;         /*像素时钟(皮秒)*/
    __u32 left_margin;      /*行切换：从同步到绘图之间的延迟*/
    __u32 right_margin;     /*行切换：从绘图到同步之间的延迟*/
    __u32 upper_margin;     /*帧切换：从同步到绘图之间的延迟*/
    __u32 lower_margin;     /*帧切换：从绘图到同步之间的延迟*/
    __u32 hsync_len;        /*水平同步的长度*/
    __u32 vsync_len;        /*垂直同步的长度*/
    __u32 sync;
    __u32 vmode;
    __u32 rotate;           /*顺时针旋转的角度*/
    __u32 reserved[5];      /*保留*/
};
```

看下图可能会对 fb_var_screeninfo 中涉及的时序更清楚一些了。




```c
struct fb_cmap {
    __u32 start;          /*第 1 个元素入口*/
    __u32 len;          /*元素数量*/
    /*R，G，B，透明度*/
    __u16 *red;
    __u16 *green;
    __u16 *blue;
    __u16 *transp;
};
```

```c
struct fb_bitfield {
    __u32 offset;       /*位域的偏移*/
    __u32 length;       /*位域的长度*/
    __u32 msb_right;    /*!=0 MSB 在右边*/
};
```

```c
struct fb_ops {
    struct module *owner;
    /*打开/释放*/
    int (*fb_open)(struct fb_info *info, int user);
    int (*fb_release)(struct fb_info *info, int user);
    /*对于非线性布局的/常规内存映射无法工作的帧缓冲设备需要*/
    ssize_t (*fb_read)(struct fb_info *info, char __user *buf,
```

```
8.              size_t count, loff_t *ppos);
9.      ssize_t (*fb_write)(struct fb_info *info, const char __user *buf,
10.              size_t count, loff_t *ppos);
11.     /*检测可变参数，并调整到支持的值*/
12.     int (*fb_check_var)(struct fb_var_screeninfo *var, struct fb_info *info);
13.     /*根据 info->var 设置 video 模式*/
14.     int (*fb_set_par)(struct fb_info *info);
15.     /*设置 color 寄存器*/
16.     int (*fb_setcolreg)(unsigned regno, unsigned red, unsigned green,
17.              unsigned blue, unsigned transp, struct fb_info *info);
18.     /*批量设置 color 寄存器，设置颜色表*/
19.     int (*fb_setcmap)(struct fb_cmap *cmap, struct fb_info *info);
20.     /*显示空白*/
21.     int (*fb_blank)(int blank, struct fb_info *info);
22.     /*pan 显示*/
23.     int (*fb_pan_display)(struct fb_var_screeninfo *var, struct fb_info *info);
24.     /*矩形填充*/
25.     void (*fb_fillrect) (struct fb_info *info, const struct fb_fillrect *rect);
26.     /*数据复制*/
27.     void (*fb_copyarea) (struct fb_info *info, const struct fb_copyarea *region);
28.     /*图形填充*/
29.     void (*fb_imageblit) (struct fb_info *info, const struct fb_image *image);
30.     /*绘制光标*/
31.     int (*fb_cursor) (struct fb_info *info, struct fb_cursor *cursor);
32.     /*旋转显示*/
33.     void (*fb_rotate)(struct fb_info *info, int angle);
34.     /*等待 blit 空闲*/
35.     int (*fb_sync)(struct fb_info *info);
36.     /*fb 特定的 ioctl*/
37.     int (*fb_ioctl)(struct fb_info *info, unsigned int cmd,
38.             unsigned long arg);
39.     /*处理 32 位的 compat ioctl*/
40.     int (*fb_compat_ioctl)(struct fb_info *info, unsigned cmd,
41.             unsigned long arg);
42.     /*fb 特定的 mmap*/
43.     int (*fb_mmap)(struct fb_info *info, struct vm_area_struct *vma);
44.     /*保存目前的硬件状态*/
45.     void (*fb_save_state)(struct fb_info *info);
46.     /*恢复被保存的硬件状态*/
47.     void (*fb_restore_state)(struct fb_info *info);
48.     /**/
49.     void (*fb_get_caps)(struct fb_info *info, struct fb_blit_caps *caps,
50.              struct fb_var_screeninfo *var);
51. };
```

```
1.  struct fb_info {
2.      int node;
3.      int flags;
4.      struct mutex lock;      /*用于 open/release/ioctl 的锁*/
5.      struct fb_var_screeninfo var;   /*可变参数*/
6.      struct fb_fix_screeninfo fix;   /*固定参数*/
7.      struct fb_monspecs monspecs;    /*显示器标准*/
8.      struct work_struct queue;   /*帧缓冲事件队列*/
9.      struct fb_pixmap pixmap;    /*图像硬件 mapper*/
10.     struct fb_pixmap sprite;    /*光标硬件 mapper*/
11.     struct fb_cmap cmap;        /*目前的颜色表*/
12.     struct list_head modelist;
13.     struct fb_videomode *mode;  /*目前的 video 模式*/
14.
15.#ifdef CONFIG_FB_BACKLIGHT
16.     struct backlight_device *bl_dev; /*对应的背光设备*/
17.     struct mutex bl_curve_mutex;    /*背光调整*/
18.     u8 bl_curve[FB_BACKLIGHT_LEVELS];
19.#endif
20.#ifdef CONFIG_FB_DEFERRED_IO
21.     struct delayed_work deferred_work;
22.     struct fb_deferred_io *fbdefio;
23.#endif
24.
25.     struct fb_ops *fbops;       /*帧缓冲操作*/
26.     struct device *device;      /*父设备*/
27.     struct device *dev;     /*fb 设备*/
28.     int class_flag;             /*私有 sysfs 标志*/
29.#ifdef CONFIG_FB_TILEBLITTING
30.     struct fb_tile_ops *tileops;    /*图块 blitting*/
31.#endif
32.     char __iomem *screen_base;  /*虚拟基地址*/
33.     unsigned long screen_size;  /*ioremapped 的虚拟内存大小*/
34.     void *pseudo_palette;       /*伪 16 色颜色表*/
35.#define FBINFO_STATE_RUNNING    0
36.#define FBINFO_STATE_SUSPENDED  1
37.     u32 state;          /*硬件状态，如挂起*/
38.     void *fbcon_par;
39.     void *par;
40. };
```