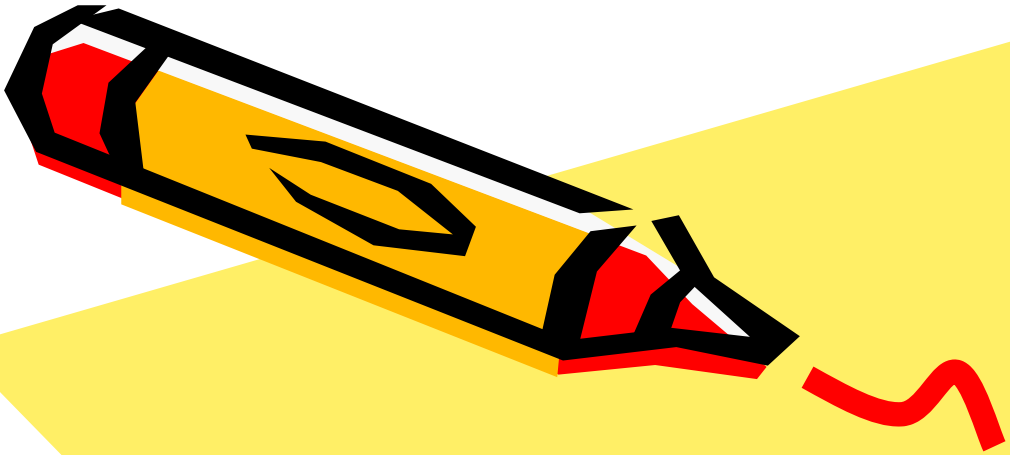
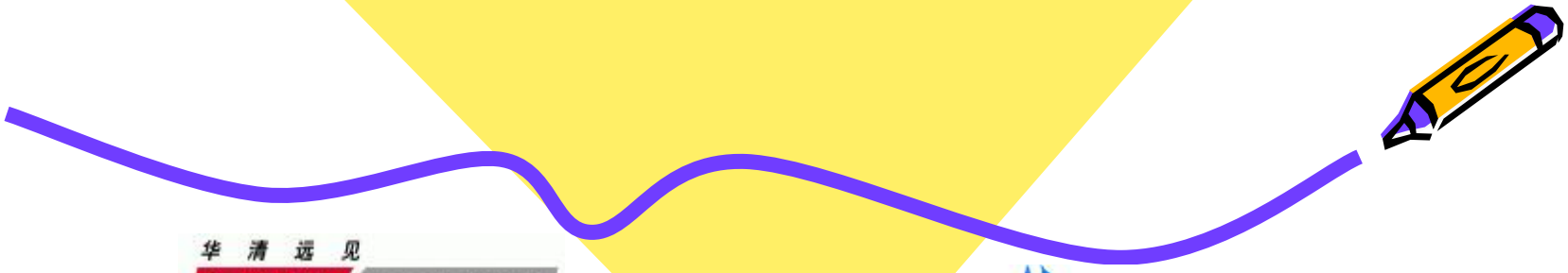


Android驱动开发入门及手机案例开发分析

华清远见



华清远见
FARSIGHT
嵌入式培训专家

嵌入式学院
华清远见旗下品牌

主要内容：

嵌入式学院
华清远见旗下品牌

1、Linux驱动开发入门
2、Android系统对Linux驱动的优化与调整
3、Android电源管理框架，策略和应用
4、基于PXA310上Andorid手机案例典型驱动开发

Linux驱动开发入门:

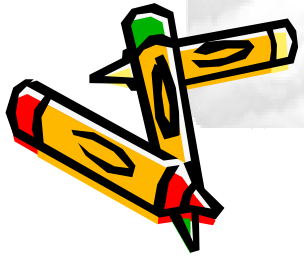
嵌入式学院
华清远见旗下品牌

1、Linux的字符设备驱动/块设备驱动/网络设备驱动

1、动手实践是最好的学习方法:

DIY一块嵌入式系统开发板(如SC2410开发板)??

买一块嵌入式系统开发板??



Android系统对Linux驱动的优化与调整:

Android内核特点:

Android比起Linux的7处增强的地方

Alarm

Ashmem/pmem

使得进程间能够共享大块的内存(如图标)

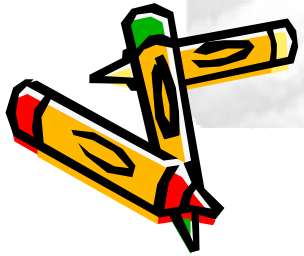
Ashmem为内核提供了一种回收这些使用完的共享内存块的办法

Binder

解决: 标准Linux进程间通信会花费许多开销,并有安全漏洞的问题

power management

提供更多的电源管理策略; 使用唤醒锁来管理电源



Android系统对Linux驱动的优化与调整:

Android内核特点:

Android比起Linux的7处增强的地方

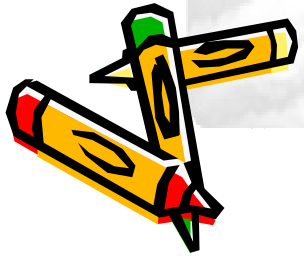
low memory killer

当内存不够的时候，该策略会试图结束一个进程

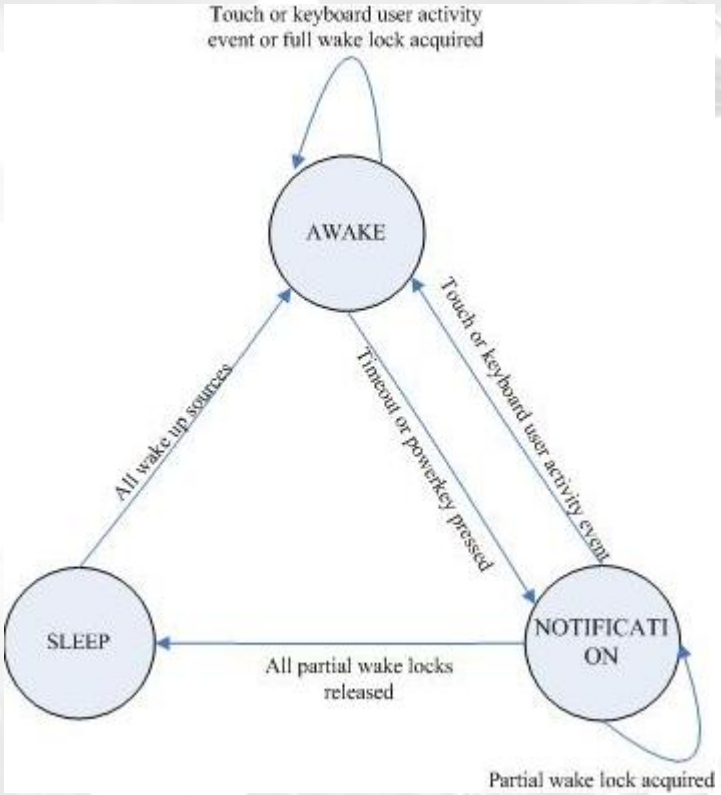
kernel debugger

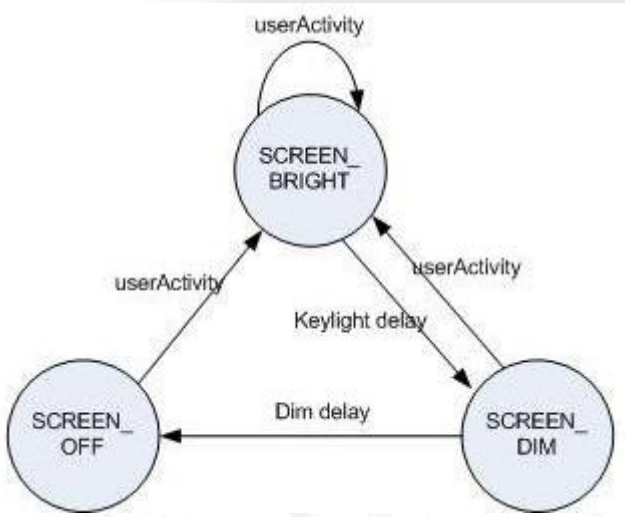
Logger

使得调试信息可以输入到一个内存块中

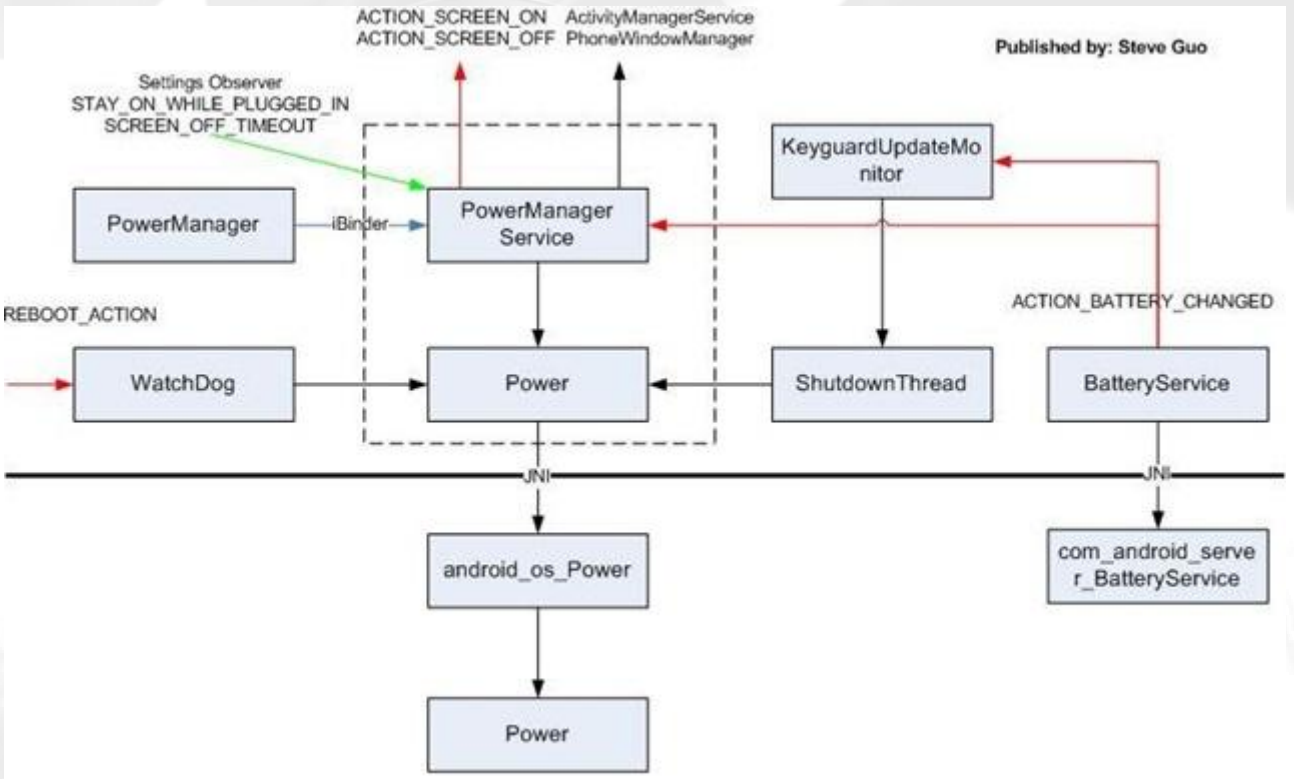


Android电源管理框架，策略和应用:





Android电源管理框架，策略和应用:



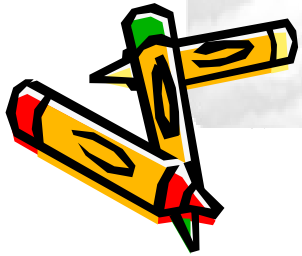
1、电源管理驱动:

Android 电源驱动会在/sys目录下创建子目录android_power并生成相应的属性文件,通过attribute的show/store来访问和设置power state/request.

```
# ls /sys/android_power
state
request_state
acquire_full_wake_lock
acquire_partial_wake_lock
release_wake_lock
```

```
# cat state
0-1-1
# cat request_state
wake
# android_power: sleep (0->2) at 157660949707

# cat request_state
standby
#
```



2、android HAL层对电源控制的操作:

hardware\libhardware\power\power.c

```
const char * const OLD_PATHS[] =
{
    "/sys/android_power/acquire_partial_wake_lock",
    "/sys/android_power/release_wake_lock",
    "/sys/android_power/request_state"
};
```

```
# cat request_state
standby
# echo wake > request_state
android_power: wakeup (2->0) at 27002401733
# android_power_suspend: done
cat request_state
wake
# echo standby > request_state
android_power: sleep (0->2) at 285930358886
# cat request_state
standby
```

2、usb驱动:

Google没有使用原来的那套gadget驱动架构(file_storage.c),而是参考file_storage.c实现了一个新的模型--- composite模型:

composite.c	// 实现android下usb管理的框架模型
android.c	// 实现具体的usb功能管理
f_mass_storage.c	// 实现优盘功能
f_adb.c	// 实现adb功能

1)该框架下,有三个设备: composite设备, 优盘设备, adb设备

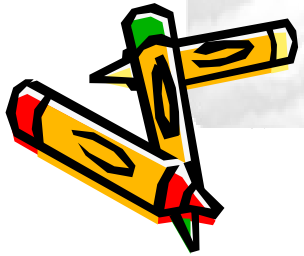
2)枚举时,首先枚举composite设备,再枚举优盘设备,最后枚举adb设备

3)composite设备被枚举时

a)在获取DEVICE描述符时,将VID,PID上报给host

b)host第一次请求CONFIG描述符时,composite设备告诉host,它有一个CONFIG,两个interface(即两个功能),以及告诉host自己使用的端点IN和端点OUT的地址

c) host会根据interface的个数决定枚举的次数(这对应着优盘枚举和adb枚举)

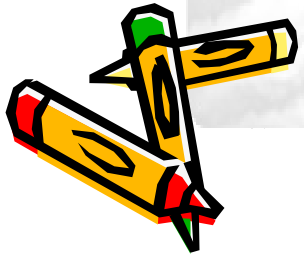


3、键盘驱动:

1)keyboard驱动要通过input_register_device()调用注册成标准的输入设备/dev/input/event0

驱动上报的按键值要和Android系统里面的/system/usr/keylayout/*.kl文件里面的记录一致,否则会导致android系统不能正确识别按键消息

```
key 232   DPAD_CENTER          WAKE_DROPPED
key 108   DPAD_DOWN            WAKE_DROPPED
key 103   DPAD_UP              WAKE_DROPPED
key 102   HOME                 WAKE
key 105   DPAD_LEFT            WAKE_DROPPED
key 106   DPAD_RIGHT           WAKE_DROPPED
```



3、键盘驱动:

2) Linux输入系统的任何一次事件通知包含如下三个信息:

事件类型 + 事件码 + 事件值(键值)

=>这3个信息的作用如下,举例说明:

对键盘输入设备来说 :事件类型EV_KEY,表明是键盘送上来的数据

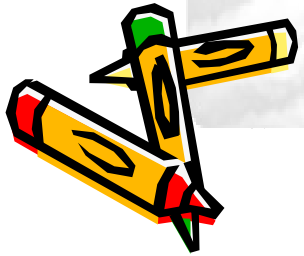
事件码KEY_CAMERA表明是键盘上的camera键被操作了

事件值表示该camera键是按下还是松开了

对触摸屏输入设备来说 :事件类型EV_ABS,表明是触摸屏送上来的绝对座标值数据

事件码ABS_X表明是触摸屏当前点的x座标

事件值表示该当前点的x座标的具体绝对值



3、键盘驱动:

3)android启动时会检测输入设备:

I/EventHub(690): New device: path=/dev/input/event2 name=ADS784x Touchscreen id=0x10000 (of 0x1) index=1 fd=44 classes=0x2

E/HAL (690): load: module=/system/lib/hw/sensors.default.so error=Cannot find library

D/SensorManager(690): found sensor: null, handle=0

I/EventHub(690): New device: path=/dev/input/event1 name=gpio-keys id=0x10001 (of 0x2) index=2 fd=46 classes=0x0

I/SystemServer(690): Starting Bluetooth Service.

I/EventHub(690): New device: path=/dev/input/event0 name=omap_twl4030keypad id=0x10002 (of 0x3) index=3 fd=47 classes=0x1

// A:

I/EventHub(690): New keyboard: publicID=65538 device->id=65538 devname='omap_twl4030keypad' propName='hw.keyboards.65538.devname' keylayout='/system/usr/keylayout/qwerty.kl'

嵌入式学院
华清远见旗下品牌

好消息：

华清深圳将于xx月开展ANDROID开发班培训

共六天，包括：

3天应用开发和3天ANDROID的系统和驱动开发

具体内容敬请关注华清远见网站:

http://www.farsight.com.cn

Q&A

谢谢！