

```
20. }
···
                       21.
                       22. static int led device open(const struct hw module t* module, const char* name,
                               struct hw device t** device)
                       24. {
                       25.
                             struct led control device t *dev;
                       26.
                       27.
                            dev = ( struct led control device t *)malloc( sizeof (*dev));
                       28.
                            memset(dev, 0, sizeof (*dev));
                       29.
                            dev->common.tag = HARDWARE DEVICE TAG;
                       30.
                       31.
                            dev->common.version = 0;
                       32.
                            dev->common.module = module;
                       33.
                            dev->common.close = led device close;
                       34.
                       35.
                            dev->set on = led on;
                       36.
                            dev->set off = led off;
                       37.
                       38.
                            *device = &dev->common:
                       39.
                       40. success:
                       41.
                             return 0:
                       42.
                       43.
                       44. static struct hw module methods t led module methods = {
                       45.
                            open: led_device_open
                       46. }:
                       47.
                       48. const struct led_module_t HAL_MODULE_INFO_SYM = {
                       49.
                            common: {
                       50.
                              tag: HARDWARE MODULE TAG,
                       51.
                              version major: 1,
                       52.
                              version minor: 0,
                       53.
                              id: LED_HARDWARE_MODULE_ID,
                       54.
                              name: "Sample LED Stub".
                       55.
                               author: "The Mokoid Open Source Project",
ďЪ
                       56.
                              methods: &led_module_methods,
0
                       57. }
                       58.
                           /* supporting APIs go here */
                      我在前面关于HAL技术的文章中已经介绍了如何写HAL stub,需要注意的只有hw_module_t和hw_device_t
···
                      这两个数据结构,这里就不复述了。
                      下面看看JNI层代码:

    struct led control device t *sLedDevice = NULL;

                        3. static jboolean mokoid_setOn(JNIEnv* env, jobject thiz, jint led)
                        4. {
                        5.
                            LOGI("LedService JNI: mokoid_setOn() is invoked.");
                        6.
                        7.
                             if (sLedDevice == NULL) {
                              LOGI("LedService JNI: sLedDevice was not fetched correctly.");
```



z516518/article/details/6893200)

₩ 989

```
在线课程
                                     (http://www.baidu.com/cb.php?c=IgF_pyfqnHmknjmsnjD0IZ0qnfK9ujYzP1mznWR10Aw-
 Bythan全核正程师15HR1rjfkn100T1dWm1f4ujT4uWb3uH6dPHNW0AwY5HDdnHc4njD4PHn0lgF 5y9YlZ0lQzq-
(http://www.baidu.com/cb
uZR8mLPpUB48ugfElAqspynETZ-
恒額? 美球
 (http://www.baidu.com/cb.php?c=IgF_pyfqnHmknjmsnjc0IZ0qnfK9ujYzP1mznWR10Aw-
                                   T0TAq15H
THE STATE OF THE S
                                  /http://www.baidu.com/cb.php?c=lgF_pyfqnHmknjmsnjn0lZ0qnfk的磁準就mznWR10Aw-
Tf0TAQ15H
  809)
112R8mLPbUB48ugfEIAq
  SDVMETVNBn6KzujYk0AF
V5H00TZcgn0KdpyfgnH
中下CMR基果的Get和Rost请求深入分析
 HTTP://Blageseln.net/5516518/article/detai
 19/69/69/29/645HcsP6KWT
(m) (2010 YrHT)
dialog,activity 屏蔽Home键详解 (http://blo
g.csdn.net/z516518/article/details/68931
70)
1184
Android JNI调用 - 文件操作 (http://blog.c
sdn.net/z516518/article/details/6868487)
1009
Android Socket编程 (http://blog.csdn.net/
```

0

<u>...</u>

ಹ

```
return -1;
10. } else {
11.
        return sLedDevice->set_on(sLedDevice, led);
12. }
13. }
14.
15. static jboolean mokoid_setOff(JNIEnv* env, jobject thiz, jint led)
16. {
17. LOGI("LedService JNI: mokoid_setOff() is invoked.");
18.
                                                                                                                                                                 \triangle
19.
                                                                                                                                                               内容举报
20.
    if (sLedDevice == NULL) {
21.
       LOGI("LedService JNI: sLedDevice was not fetched correctly.");
                                                                                                                                                                 TOP
22.
                                                                                                                                                               返回顶部
23. } else {
24.
        return sLedDevice->set_off(sLedDevice, led);
25. }
26. }
27.
28. /** helper APIs */
29. static inline int led_control_open( const struct hw_module_t* module,
        struct led_control_device_t** device) {
31.
      return module->methods->open(module,
          LED_HARDWARE_MODULE_ID, ( struct hw_device_t**)device);
32.
33. }
34.
35. static jboolean mokoid_init(JNIEnv *env, jclass clazz)
36. {
37. led_module_t* module;
38.
39.
    if (hw get module(LED HARDWARE MODULE ID, (const hw module t**)&module) == 0) {
       LOGI("LedService JNI: LED Stub found.");
41.
        if (led control open(&module->common, &sLedDevice) == 0) {
42.
          LOGI("LedService JNI: Got Stub operations.");
43.
           return 0:
44.
45.
47. LOGE("LedService JNI: Get Stub operations failed.");
48.
      return -1;
49. }
50.
51. static const JNINativeMethod gMethods[] = {
52. { "_init", "()Z", ( void *)mokoid_init },
53. { " set on", "(I)Z", ( void *)mokoid setOn },
54. { "_set_off", "(I)Z", ( void *)mokoid_setOff },
55. };
57. int register_mokoid_server_LedService(JNIEnv* env) {
58. static const char* const kClassName =
                                                                                                                                                                 ⚠
       "com/mokoid/LedClient/LedClient";
                                                                                                                                                               内容举报
```

0

<u>...</u>

ಹ

```
60. jclass clazz;
 61.
 62. /* look up the class */
                                                                                                                                                返回顶部
 63. clazz = env->FindClass(kClassName);
 64. if (clazz == NULL) {
 65.
       LOGE("Can't find class %s\n", kClassName);
 67. }
 68.
 69. /* register all the methods */
     if (env->RegisterNatives(clazz, gMethods,
 71.
           sizeof (gMethods) / sizeof (gMethods[0])) != JNI OK)
 72. {
 73.
        LOGE("Failed registering methods for %s\n", kClassName);
 74.
        return -1;
 75. }
 76.
 77. /* fill out the rest of the ID cache */
 78.
      return 0:
 79. }
 80.
 81. extern "C" jint JNI_OnLoad(JavaVM* vm, void * reserved)
 82. {
 83. JNIEnv* env = NULL;
 84. jint result = -1;
 85.
 86.
      if (vm->GetEnv(( void **) &env, JNI_VERSION_1_4) != JNI_OK) {
 87.
       LOGE("GetEnv failed!");
 88.
        return result;
 89. }
 90. LOG ASSERT(env, "Could not retrieve the env!");
 91.
 92.
     register mokoid server LedService(env);
 93.
      return JNI VERSION 1 4;
 95. }
上面的Jni代码首先通过hw_get_module得到HAL stub, open以后就可以直接使用HAL stub中定义的接口。
这里还需要注意JNI OnLoad这个函数,当Jni库被App load的时候,该函数将会自动被调用,所以我们在这
里实现了注册Led Service的操作,也就是说把C/C++的接口映射到Java中去,这样在Java APP中就可以使
                                                                                                                                                 \triangle
用该接口了。在register mokoid server LedService中,我们需要注意kclassname指定了需要调用该Jni库
                                                                                                                                                内容举报
的Java APP类 - com.mokoid.LedClient.LedClient , 也就是说该Jni库只能提供给该Java程序使用。
最后是应用程序代码:
                                                                                                                                                 TOP
  1. public class LedClient extends Activity {
                                                                                                                                                返回顶部
  2.
  3.
      static {
        System.load("/system/lib/libmokoid_runtime.so");
  5. }
  6.
  7. @Override
  8. public void onCreate(Bundle savedInstanceState) {
        super .onCreate(savedInstanceState);
```

```
10.
                 11.
                       // Call an API on the library.
                 12.
                        init();
ß
                 13.
                       set on(1);
                 14.
                       set off(2);
                 15.
                 16.
                        TextView tv = new TextView( this ):
                 17.
                        tv.setText("LED 1 is on. LED 2 is off.");
\odot
                 18.
                        setContentView(tv);
                 19. }
ಹ
                     private static native boolean init();
                      private static native boolean set on( int led);
                 22.
                      private static native boolean _set_off( int led);
                 23. }
                 上面使用System.load来装载Jni库,当然我们也可以使用System.loadLibrary来装载,他们的唯一区别就是
                 前者需要指定完整的路径,后者会在系统路径上(比如/system/lib/)查找库。装载完该Jni库后,就可以使
                 用映射后的接口了(init, set on, set off)。
                 上面这种HAL的实现方式比较简单,但是也存在一个很大的问题,就是Jni库只能提供给某一个特定的Java
                 使用,如何克服这个问题?我们可以在APP和Jni之间加一层Java service,该Jni提供给Java service使用,
                 而所有的APP利用该service来使用Jni提供的接口。这样的话,在应用程序层,就不需要关心Jni是如何实现
                的了。下一篇我们会介绍这种方法。
                 本文出自 "Mobile and Linux Deve.. (http://buaadallas.blog.51cto.com/)" 博客,请务必保留此出
                 处http://buaadallas.blog.51cto.com/399160/384622 (http://buaadallas.blog.51cto.com/399160/384622)
                     发表你的评论
                (http://my.csdn.net/weixin 35068028)
                                               相关文章推荐
               Android HAL实现的三种方式(1) - 基于JNI的简单HAL设计 (http://blog.csdn.net/mdx2007241...
               现在在Android上的HAL开发总的来说还是随意性比较大,Android也并没有规范好一个具体的框架,下面我将根据Jollen的Mo
               koid工程,自己做了一些改动,分别给大家介绍一下三种实现方式。...
```

Android HAL实现的三种方式(1) - 基于JNI的简单HAL设计 (http://blog.csdn.net/cjok3762404...
现在在Android上的HAL开发总的来说还是随意性比较大,Android也并没有规范好一个具体的框架,下面我将根据Jollen的Mo

koid工程,自己做了一些改动,分别给大家介绍一下三种实现方式。 ...

ga cjok376240497 (http://blog.csdn.net/cjok376240497) 2012年04月09日 17:02 Q 856

AI 专业人才缺口上百万 年薪 80 万远超同行??

内容举报

返回顶部

http://blog.csdn.net/z516518/article/details/6805156

ß

 \odot

ಹ



이 목표사이에서 그러가 가게 안 가셨습니다 . .

就目前来看,国内 AI 人才缺乏且经验不足,为争抢优秀人才,企业背后的暗战早已打响。作为正在谋 求一份好工作我,又该如何抉择....

(http://www.baidu.com/cb.php?c=IgF_pyfqnHmknjnvPjn0IZ0qnfK9ujYzP1ndPWb10Aw-

5Hc3rHnYnHb0TAg15HfLPWRznib0T1dWni01PWbzuhRYm19BPWcs0AwY5HDdnHc4niD4PHn0IgF 5v9YIZ0IOzg-

uZR8mLPbUB48ugfEIAqspynEmybz5LNYUNq1ULNzmvRqmhkEu1Ds0ZFb5HD0mhYqn0KsTWYs0ZNGujYkPHTYn1mk0AqGujYknWb3rjDY0APGujYLnWm4n1c0ULi85H00TZbqnW0v0APzm1Y1rjcsP6)

Android HAL实现的三种方式(1) - 基于JNI的简单HAL设计 (http://blog.csdn.net/shentong1/a...

转载自 http://buaadallas.blog.51cto.com/399160/384622



Android HAL实现1——Java应用程序直接调用JNI库 (http://blog.csdn.net/manshq163com/...

先介绍最简单的一种实现方式 - Java应用程序直接调用JNI库。 由于JNI技术的存在,在Android中,java程序能够很好的调 用C/C++库。我们这里设计一个简单的HAL,一共只...



⚠ 内容举报

TOP 返回顶部

Android HAL实现的三种方式(3) - 基于Manager的HAL设计 (http://blog.csdn.net/cjok376240...

在上文中我们实现了利用Service来设计HAL,但是根据Android的框架,一般应用程序都不直接和Service打交到,中间需要 经过一个Manager层。 我们在这里也按照Android的框...



AI 工程师职业指南

我们请来商汤、杜邦、声智、希为、58同城、爱因互动、中科视拓、鲁朗软件等公司 AI 技术一线的专 家,请他们从实践的角度来解析 AI 领域各技术岗位的合格工程师都是怎样炼成的。

(http://www.baidu.com/cb.php?c=IgF pyfqnHmknjfzrj00IZ0qnfK9ujYzP1f4Pjnd0Aw-

rA7Wuj0YmhP9PARvujmYmH0vm1qdIAdxTvqdThP-

5HDknWF9mhkEusKzujYk0AFV5H00TZcqn0KdpyfqnHRLPjnvnfKEpyfqnHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqnWn1rjT)

王家林最受欢迎的一站式云计算大数据和移动互联网解决方案课程 V1之Android架构设计和实...

如何理解Android架构设计的初心并开发出搭载Android系统并且具备深度定制和软硬整合能力特色产品,是本课程解决的问 题。 课程以Android的五大核心: HAL、Binder、NativeSe...

Rocky_wangjialin (http://blog.csdn.net/Rocky_wangjialin) 2014年08月11日 21:04 □ 1613

Android HAL实现的三种方式(3) - 基于Manager的HAL设计 (http://blog.csdn.net/RationalGo...

在上文中我们实现了利用Service来设计HAL,但是根据Android的框架,一般应用程序都不直接和Service打交到,中间需要 经过一个Manager层。 我们在这里也按照Android的框...

凸

°

...

在上一篇文章中,我介绍了一种应用程序直接调用JNI库的HAL设计方法,该方法虽然简单,但是不符合Android的框架结构,下面我们介绍一种通过Service提供接口给应用程序的设计方法,结构如下:...

mdx20072419 (http://blog.csdn.net/mdx20072419) 2012年04月10日 12:57 贝517

Android之 看"马达"如何贯通Android系统 (从硬件设计 --> 驱动 --> HAL --> JNI --> Framewo...

在Android 2.3(Gingerbread) 系统的时候,我写过一篇关于"Android 震动马达系统"的文章,当时的Linux内核还是2.6版本的。写那篇文章的目的,是想彻底的了解从硬件到驱动...

深入浅出 - Android系统移植与平台开发(十) - led HAL简单设计案例分析 (http://blog.csdn....

通过前两节HAL框架分析和JNI概述,我们对Android提供的Stub HAL有了比较详细的了解了,下面我们来看下led的实例,写 驱动点亮led灯,就如同写程序,学语言打印HelloWorld一样,...

深入浅出 - Android系统移植与平台开发(十) - led HAL简单设计案例分析 (http://blog.csdn....

作者:唐老师,华清远见嵌入式学院讲师。 通过前两节HAL框架分析和JNI概述,我们对Android提供的Stub HAL有了比较详细的了解了,下面我们来看下led的实例,写驱动点亮led灯,就如同写程…

each farsight2009 (http://blog.csdn.net/farsight2009) 2013年10月28日 11:35 2975

深入浅出 - Android系统移植与平台开发(十) - led HAL简单设计案例分析 (http://blog.csdn....

通过前两节HAL框架分析和JNI概述,我们对Android提供的Stub HAL有了比较详细的了解了,下面我们来看下led的实例,写 驱动点亮led灯,就如同写程序,学语言打印HelloWorld一样,...

[| lqxandroid2012 (http://blog.csdn.net/lqxandroid2012) 2015年08月12日 21:04 2437

深入浅出 - Android系统移植与平台开发(十) - led HAL简单设计案例分析 (http://blog.csdn....

版权声明:本文为博主原创文章,未经博主允许不得转载。 目录(?)[+] 通过前两节HAL框架分析和JNI概述,我们对Android提供的Stub HAL...

王家林最受欢迎的一站式云计算大数据和移动互联网解决方案课程 V1 (20140809) 之Android...

掌握Android从底层开发到框架整合技术到上层App开发及HTML5的全部技术; 一次彻底的Android架构、思想和实战技术的洗礼; 彻底掌握Andorid HAL、Android Runti...

Rocky_wangjialin (http://blog.csdn.net/Rocky_wangjialin) 2014年08月11日 20:41 1825

Android LED灯5层 (app,framework,jni,hal,drivers)源码,可扩展GPIO (h...

⚠
内容举报

心 返回顶部

⚠
内容举报

企 返回顶部

 \odot

(httn://download.c	2014年12月29日 13:56	798KB	下载				
Docx (http://download.c	android从HAL , ji 2013年09月12日 08:41		p 学习心 行	寻,绝对原	创 (http://d	ownload.csd	n.net
Android Camera	2013年09月12日 08:41 a Hal 的初步实现1(net/lqf785	435771/arti	icle/details/6	9652
参考http://hi.baidu.com	n/aokikyon/blog/item/ee9	339f3e1b23	3fc60b46e0	47.html 的设i	十转载:http://bl	og.csdn.net/	
(ht	tp://blog.csdn.net/lqf7854	435771)	2011年11月	13日 09:42	<u>1204</u>		
g照片。 1、修改你的E	、, 这个摄像头返回JPEG BoardConfig.mk ttp://blog.csdn.net/zhimil					() ,但是可以直接	€take jpe
在Ubuntu为And	roid 硬件抽象层 (H	IAL)模均	块编写JN	方法提供J	ava访问硬例	牛服务接口 (h	ttp://
]介绍了如何为Android系 设口。实现这两者的目的是					实现内核驱动程序	和在用户
ydt_lwj (http://blo	g.csdn.net/ydt_lwj) 20	12年07月09	9日 11:53	□269			
基于Android2.3.	5系统:JNI与HAL3	实例解析[—] (http	://blog.cs	dn.net/wav	emcu/article/	detai
Android系统下的JNI的 代码进行交互。JNI 是]全称是:Java Native Int 本地编程接口,它…	erface (JNI)),JNI标准	是java平台的-	一部分,它允许	Java代码和其他证	言写的

🎒 wavemcu (http://blog.csdn.net/wavemcu) 2015年01月25日 19:45 🕮 1267

⚠
内容举报

(元) 返回顶部