

Android硬件抽象层

罗升阳

<http://weibo.com/shengyangluo>

<http://blog.csdn.net/luoshengyang>

About Me

- 《老罗的Android之旅》 博客作者
- 《Android系统源代码情景分析》 书籍作者
- 博客: <http://blog.csdn.net/Luoshengyang>
- 微博: <http://weibo.com/shengyangluo>

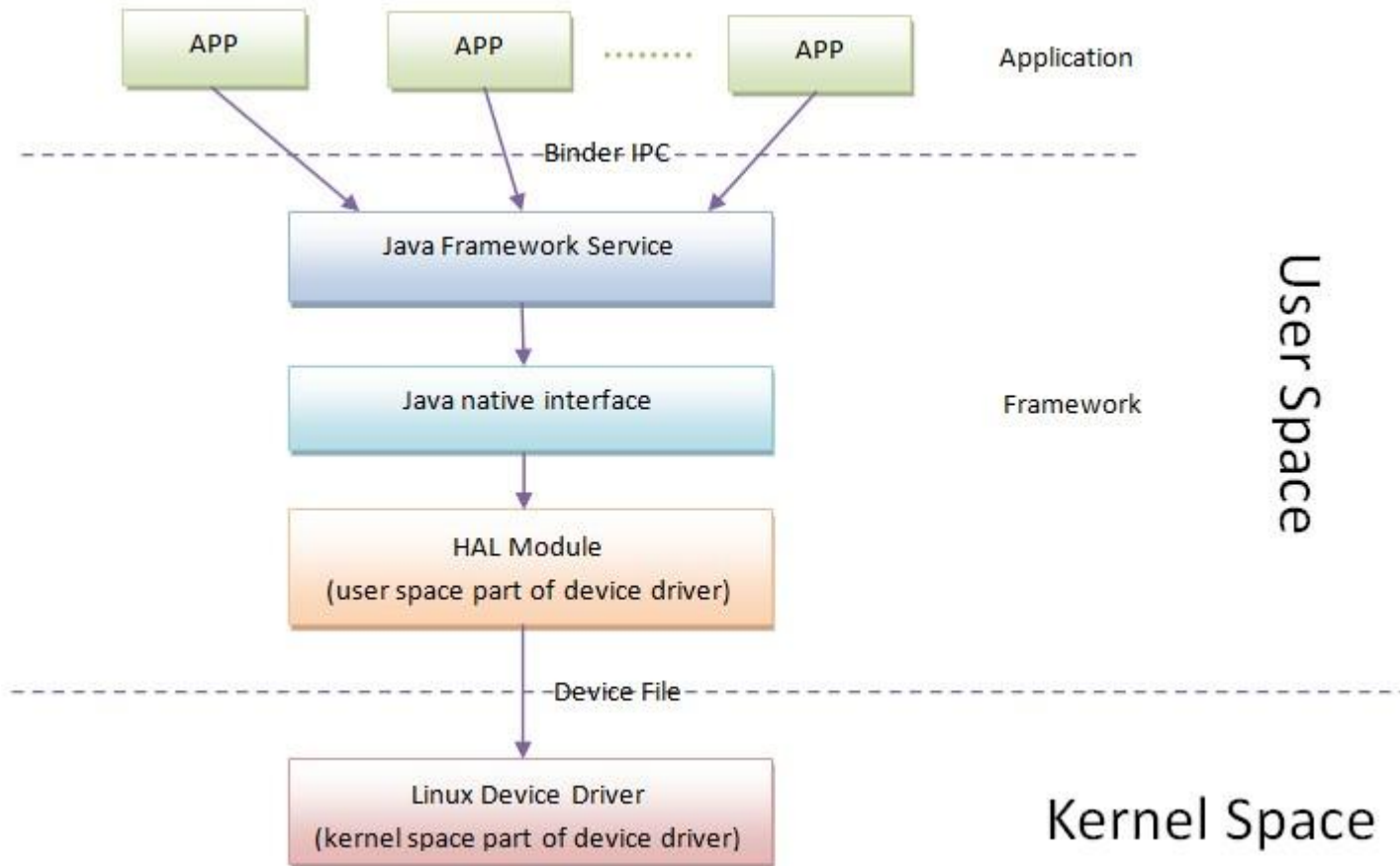
Agenda

- Android硬件抽象层概述
- Android硬件驱动程序开发
- Android硬件驱动程序验证
- Android硬件抽象层模块开发
- Android硬件访问服务开发
- Android应用程序开发

Android硬件抽象层概述

- 设备驱动分为内核空间和用户空间两部分
 - 保护厂商利益（出发点）
 - 内核空间主要负责硬件访问逻辑（GPL）
 - 用户空间主要负责参数和访问流程控制（Apache License）
- 用户空间部分设备驱动即为HAL Module
 - HAL Module通过设备文件访问内核空间部分设备驱动
- 系统服务通过HAL Module对硬件进行管理
 - 系统服务通过JNI访问HAL Module
- 应用程序通过系统服务对硬件进行访问
 - 应用程序通过Binder IPC访问系统服务

Android硬件抽象层概述(续)



Android硬件驱动程序开发

- 与传统的Linux硬件驱动程序开发是一样的
 - 实现驱动程序
 - 包括源代码文件、编译脚本文件、编译配置文件
 - 提供proc、devfs和sysfs三种文件系统访问接口
 - 修改根Kconfig文件
 - 修改根Makefile文件
 - 编译驱动程序

Android硬件驱动程序验证

- 验证proc文件系统访问接口
 - 通过cat和echo命令验证
- 验证sysfs文件系统访问接口
 - 通过cat和echo命令验证
- 验证devfs文件系统访问接口
 - 编写C程序通过open、read和write系统调用验证

Android硬件抽象层模块开发

- 模块文件命名规范

```
/**
 * There are a set of variant filename for modules. The form of the filename
 * is "<MODULE_ID>.variant.so" so for the led module the Dream variants
 * of base "ro.product.board", "ro.board.platform" and "ro.arch" would be:
 *
 * led.trout.so
 * led.msm7k.so
 * led.ARMV6.so
 * led.default.so
 */

static const char *variant_keys[] = {
    "ro.hardware", /* This goes first so that it can pick up a different
                    file on the emulator. */
    "ro.product.board",
    "ro.board.platform",
    "ro.arch"
};
```


Android硬件抽象层模块开发(续)

- 定义模块ID
- 定义设备ID
- 定义模块结构体
 - 第一个成员变量必须是标准的hw_module_t结构体
 - 相当于是定义一个hw_module_t子类
- 定义设备结构体
 - 第一个成员变量必须是标准的hw_device_t结构体
 - 相当于是定义一个hw_device_t子类
- 定义符号HAL_MODULE_INFO_SYM，类型为自定义的模块结构体
- 实现设备打开接口（必须）
- 实现设备关闭接口（必须）
- 实现设备访问接口（可选）

Android硬件抽象层模块开发(续)

- 模块加载过程：hw_get_module
 - 依次在/system/lib/hw和/vendor/lib/hw目录中检查是否存在相应的“<MODULE_ID>.variant.so”文件。其中，variant分别等于属性“ro.hardware”、“ro.product.board”、“ro.board.platform”和“ro.arch”的值。只要其中一个存在，即停止查找。
 - 如果上述文件均不存在，则继续在/system/lib/hw目录中检查“<MODULE_ID>.variant.so”文件是否存在。
 - 调用dlopen打开上述找到的so文件。
 - 调用dlsym获得上述打开的so文件里面的符号HAL_MODULE_INFO_SYM。
 - 将符号HAL_MODULE_INFO_SYM强制转换为一个hw_moudle_t结构体。

Android硬件抽象层模块开发(续)

- 修改设备文件访问权限
 - 设备文件在默认情况下只有root用户可以访问
 - 设备文件一般是在非root用户进程中访问的
 - 修改ueventd.rc文件赋予设备文件非root用户访问权限
- 修改ueventd.rc文件的方法
 - 解压ramdisk.img文件，得到ramdisk.img.gz归档文件
 - 解除ramdisk.img.gz文件归档，得到ramdisk目录
 - 修改ramdisk目录下的ueventd.rc文件
 - 重新打包ramdisk.img镜像文件

Android硬件访问服务开发

- 定义硬件访问接口IXXX
 - 使用AIDL语言定义
 - 编译后会生成一个IXXX.Stub类
- 实现硬件访问服务XXX
 - 从IXXX.Stub类继承
 - 实现硬件访问接口IXXX
 - 通过JNI访问硬件抽象层模块
- 实现硬件访问服务XXX的JNI接口
 - 调用函数hw_get_module加载硬件抽象层模块
 - 打开硬件设备
- 启动硬件访问服务
 - 在System Server进程中创建一个XXX实例
 - 调用ServiceManager.addService接口将XXX实例注册到Service Manager中

Android应用程序开发

- 调用ServiceManager.getService接口获得硬件访问服务XXX的代理接口
- 通过代理接口访问硬件访问服务

Q&A

Thank You