

《深入解析Android 5.0系统》

图书基本信息

书名：《深入解析Android 5.0系统》

13位ISBN编号：9787115384568

出版时间：2015-4-1

作者：刘超

页数：708

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

《深入解析Android 5.0系统》

内容概要

《深入解析Android 5.0系统》详细剖析了最新Android 5.0系统主要框架的原理和具体实现。本书共24章，覆盖了Android 5.0系统中、下层重要的模块，对于每个模块都详细介绍了它们的架构、原理及代码实现等各个方面，尽量让读者知其然，又知其所以然，达到学以致用目的。主要内容为：

Android Build系统核心、Android的Bionic、系统调用的实现方法、Android的Binder、Binder应用层的核心类、Android的JNI、Android的同步和消息机制、进程间的消息传递、Android的Init进程、Android的Zygote进程、Android的资源管理、Android的SystemServer进程、Android的应用管理、Android的组件管理、Android的多用户模式、Android的图形显示系统、Android的窗口系统、Android的输入管理、Android的电源管理、Android的存储系统、Android的网络管理框架、Android的音频系统、Android的SELinux模块、Dalvik和ART虚拟机、Android的Recovery模块、Android的调试方法、内存泄露的分析、Android的自动化测试等系统的核心知识。

在书中尽可能详细地给出了代码的注释、各种属性和常量的解释，以及各种系统中使用的文件格式的介绍，希望读者能通过本书，获得对Android 5.0系统进行二次开发的能力，本书是进行系统开发人员的案头必备书。

《深入解析Android 5.0系统》面向的读者主要是进行系统开发的工程师，包括应用开发工程师、ROM开发工程师和各种使用Android作为开发平台的TV和可穿戴式设备(Wear)的开发工程师。本书也可以作为大专院校相关专业师生的学习用书及培训学校教材。

书籍目录

第1章 建立Android系统开发环境 1

1.1 安装操作系统 1

1.1.1 安装方式的选择 1

1.1.2 下载和安装Ubuntu 1

1.1.3 使用Ubuntu遇到的问题 2

1.2 安装开发包 3

1.2.1 安装JDK 1.6 3

1.2.2 安装OpenJDK 1.7 4

1.2.3 安装编译需要的开发包 4

1.3 安装一些有用的工具 4

1.3.1 安装Android SDK 4

1.3.2 安装Android Studio 4

1.3.3 安装Source Insight 5

1.3.4 安装比较工具Meld 5

1.4 下载源码 5

1.4.1 Git and Repo简介 5

1.4.2 源码版本历史 6

1.4.3 下载Android源码 7

1.4.4 下载Kernel源码 8

第2章 Android的编译环境——Build系统 10

2.1 Android Build系统核心 10

2.1.1 编译环境的建立 11

2.1.2 Build相关的环境变量 14

2.1.3 Build系统的层次关系 15

2.1.4 分析main.mk文件 17

2.1.5 Build系统的编译目标

介绍 20

2.1.6 分析config.mk文件 22

2.1.7 分析product_config.mk

文件 24

2.1.8 Android 5.0中的64位

编译 26

2.2 Android的产品配置文件 27

2.2.1 分析hammerhead的配置

文件 27

2.2.2 编译类型eng、user和

userdebug 31

2.2.3 产品的Image文件 32

2.2.4 如何加快编译速度 33

2.2.5 如何编译Android的

模拟器 34

2.3 编译Android的模块 34

2.3.1 模块编译变量简介 35

2.3.2 常用模块定义实例 36

2.3.3 预编译模块的目标定义 37

2.3.4 常用“LOCAL_”变量 39

- 2.4 Android中的签名 40
 - 2.4.1 Android应用签名方法 41
 - 2.4.2 Android系统签名介绍 43
 - 2.4.3 Android签名漏洞分析 44
- 第3章 连接Android和Linu内核的
桥梁——Android的Bionic 46
 - 3.1 Bionic简介 46
 - 3.1.1 Bionic的特性 46
 - 3.1.2 Bionic中的模块简介 49
 - 3.2 Bionic C库中的系统调用 50
 - 3.2.1 系统调用简介 50
 - 3.2.2 系统调用的实现方法 51
 - 3.3 Bionic中的内存管理函数 52
 - 3.3.1 系统调用brk和mmap 52
 - 3.3.2 内存分配器——dldmalloc
简介 53
 - 3.3.3 dldmalloc函数用法指南 54
 - 3.4 管道 57
 - 3.4.1 匿名管道PIPE和命名
管道FIFO 57
 - 3.4.2 匿名管道的使用方法 58
 - 3.5 Bionic中的线程管理函数 59
 - 3.5.1 Bionic线程函数的特性 59
 - 3.5.2 创建线程和线程的属性 59
 - 3.5.3 退出线程的方法 61
 - 3.5.4 线程本地存储TLS 62
 - 3.5.5 线程的互斥量 (Mutex)
函数 63
 - 3.5.6 线程的条件量 (Condition)
函数 65
 - 3.6 Futex同步机制 66
 - 3.6.1 Futex的系统调用 66
 - 3.6.2 Futex的用户态操作 67
 - 3.6.3 Mutex类使用Futex
实现同步 68
 - 3.7 Android的 Log模块 68
 - 3.7.1 Android Log系统的架构 69
 - 3.7.2 Log系统的接口和用法 70
 - 3.7.3 Log系统的实现分析 71
 - 3.8 可执行文件格式分析 75
 - 3.8.1 ELF格式简介 75
 - 3.8.2 ELF文件头格式 76
 - 3.8.3 程序头部表 77
 - 3.8.4 与重定位相关的“节区”的
信息——DYNAMIC段 79
 - 3.8.5 函数的重定位过程 81
 - 3.9 Bionic中的Linker模块 84
 - 3.9.1 可执行程序的装载 84
 - 3.9.2 可执行程序的初始化 85

3.9.3 Linker装载动态库	87
3.10 调试器——Ptrace和Hook API	91
3.10.1 ptrace函数简介	91
3.10.2 Hook API的原理	92
3.10.3 利用ptrace实现Hook API	93
第4章 进程间通信——Android的Binder	98
4.1 Binder简介	98
4.1.1 Binder对象定义	98
4.1.2 Binder的架构	99
4.1.3 组件Service和匿名Binder服务	100
4.1.4 Binder的层次	101
4.2 如何使用Binder	102
4.2.1 使用Binder服务	102
4.2.2 Binder的混合调用	102
4.2.3 用Java开发Binder服务	103
4.2.4 用C++开发Binder服务	104
4.3 Binder应用层的核心类	106
4.3.1 IInterface中的两个宏	106
4.3.2 Binder核心类的关系	107
4.3.3 函数asInterface的奥秘	109
4.3.4 Binder的“死亡通知”	110
4.3.5 Java层的Binder类	111
4.4 Binder的实现原理	115
4.4.1 Binder的线程模型	115
4.4.2 Binder对象的传递	119
4.4.3 分析IPCThreadState类	122
4.5 Binder驱动	126
4.5.1 应用层和驱动的消息协议	126
4.5.2 Binder驱动分析	129
4.5.3 Binder的内存共享机制	130
4.5.4 驱动的ioctl操作	131
4.5.5 Binder调用过程	133
4.5.6 处理传递的Binder对象	138
4.6 解析名称的模块——ServiceManager的作用	140
4.6.1 ServiceManager的架构	141
4.6.2 ServiceManger提供的服务	143
4.7 匿名共享内存ashmem	146
4.7.1 ashmem的作用和用法	146
4.7.2 ashmem驱动的实现原理	148
4.7.3 ashemem驱动的代码分析	149
4.7.4 进程间传递文件描述符	152
第5章 连接Java和C/C++层的	

- 关键——Android的JNI 154
- 5.1 JNI的作用 154
- 5.2 JNI用法介绍 154
 - 5.2.1 从Java到C/C++ 154
 - 5.2.2 从C/C++到Java的调用 158
- 5.3 JNI环境 160
 - 5.3.1 结构体JNIEnv 160
 - 5.3.2 JNIEnv的创建和初始化 162
 - 5.3.3 JNI中的异常处理 163
 - 5.3.4 JNI中的引用 164
 - 5.3.5 指明错误位置——“ CheckJNI ” 的作用 165
- 5.4 ART带来的JNI变化 165
 - 5.4.1 垃圾回收的影响 165
 - 5.4.2 错误处理的变化 166
 - 5.4.3 堆栈可能引发的问题 166
- 第6章 Android的同步和消息机制 167
- 6.1 原子操作 167
 - 6.1.1 Android的原子操作函数 167
 - 6.1.2 原子操作的实现原理 168
 - 6.1.3 内存屏障和编译屏障 169
- 6.2 Android native层的同步方法 171
 - 6.2.1 互斥体Mutex和自动锁Autolock 171
 - 6.2.2 解决线程同步——条件类Condition 173
- 6.3 Android Java层的同步机制 174
 - 6.3.1 同步关键字synchronized 174
 - 6.3.2 Object类在同步中的作用 175
- 6.4 Android的消息机制 176
 - 6.4.1 消息模型 177
 - 6.4.2 理解Looper类 178
 - 6.4.3 理解Handler类 180
 - 6.4.4 消息的同步——Message类的setAsynchronous()方法 181
 - 6.4.5 分析MessageQueue类 182
- 6.5 进程间的消息传递 186
 - 6.5.1 理解Messenger类 187
 - 6.5.2 建立通信通道——AsyncChannel类的作用 187
- 第7章 第一个用户进程——Android的Init进程 192
- 7.1 Init进程的初始化过程 194
 - 7.1.1 main函数的流程 194
 - 7.1.2 启动
- 。。。。。。

《深入解析Android 5.0系统》

精彩短评

- 1、可能我基础太差 看起来挺费劲的
- 2、bullshit
- 3、作者选择解析的源码不够全面，可能作者选择的是他最熟悉的吧，如果就当这本书算是源码解析方面的补漏还可以。
- 4、搞不清作者到底想不想好好写这本书，挺厚的，但看完觉得除了大框架概念之外，没什么技术营养.....疏漏就不说了

章节试读

1、《深入解析Android 5.0系统》的笔记-第192页

应该是先启动SystemServer然后由其启动SystemService。书中文字描述的时候，文字反了。

2、《深入解析Android 5.0系统》的笔记-第512页

WindowInputEventReceiver的构造方法为:public WindowInputEventReceiver(InputChannel inputChannel, Looper looper) {而不是public InputEventReceiver(InputChannel inputChannel, Looper looper) {在构造方面里面调用了super的构造方法，才是文章写的，InputEventReceiver。

3、《深入解析Android 5.0系统》的笔记-第521页

WindowManagerService中有一个TouchMode模式，这种模式下有输入焦点的Widget也无需高亮显示，这样我们点击某个按钮时按钮会高亮，这样更美观。但是当用户改用键盘操作时，则会退出TouchMode模式，这时屏幕上有焦点的Widget将会高亮显示，用来提示用户焦点所在。这段话说了之前遇到过几次的一个问题。默认情况下是在TouchMode模式，这个时候如果你想通过选中某个view，让其获得焦点都没有问题，但是并不会高亮现实。当然普通设备上没问题，如果你的系统是在一个笔记本上面，默认有实体键盘，可能期望的是选择之后高亮显示，这就在现实上没有符合预期了。一个简单的方案就是进行手工键盘时间注入，并不完美，但是比较好用。当然最开始就没有前面假设的需求则不会有这个问题存在。

《深入解析Android 5.0系统》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com