

图书基本信息

书名：<<Android核心原理与系统级应用高效开发>>

13位ISBN编号：9787121173615

10位ISBN编号：7121173611

出版时间：2012-7

出版时间：电子工业出版社

作者：韩超

页数：434

字数：717000

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：<http://www.tushu007.com>

## 内容概要

《android核心原理与系统级应用高效开发》是一本介绍android核心开发的技术图书，主要目标是让android的开发者更深入地理解android系统，让开发的过程更正确、更有效率。

本书主要由两大部分组成，第1篇自下而上介绍android核心框架层的结构，并将开发的技巧和要点蕴含于其中；第2篇偏重系统级的应用层，关注android开发过程中的重点方面。

《android核心原理与系统级应用高效开发》为android系统级别的开发者系统化而有条理性地展示了庞大的android系统的软件结构，点明开发思路；android应用程序的开发者可以通过本书加深认识，拓展开发手段；从事android硬件移植的开发者可以通过本书加深对系统全局的了解。

## 作者简介

### 韩超

中国大陆地区Linux和移动设备领域的资深工程师、架构师。

长期从事一线开发工作，兼具开发实践经验和完备的教育思维模式。

韩超是中国大陆地区较早开发Android系统的人员，曾从事系统框架、芯片移植适配和移动产品等开发领域；也曾经引领各种技术人员进入Android领域，并组织参与技术交流。

曾出版《Android系统原理及开发要点详解》、《Android系统级深入开发——移植与调试》、《Android经典应用程序开发》、《Android核心原理与系统级应用高效开发》等作品。

## 书籍目录

## 第1篇 Android核心框架层开发

## 第1章 Android系统框架

## 1.1 Android系统基础结构

## 1.2 Android的开发环境

## 1.2.1 开发环境

## 1.2.2 源代码获取

## 1.2.3 编译Android源代码

## 1.2.4 仿真器运行

## 1.2.5 使用工具

## 1.3 系统结构和代码结构

## 1.3.1 软件系统结构

## 1.3.2 源代码结构

## 第2章 编译结构和各种构建

## 2.1 Android的系统构建

## 2.1.1 编译环境要求

## 2.1.2 构建流程

## 2.1.3 环境设置

## 2.1.4 系统构建结果

## 2.2 各种部件的构建

## 2.2.1 Android.mk的语法

## 2.2.2 各种部件的构建方式

## 2.2.3 预编译内容的安装

## 2.3 Android编译的板级支持

## 2.3.1 支持一个板的工作

## 2.3.2 buildspec.mk文件的作用

## 2.3.3 vendersetup.sh和lunch命令

## 2.3.4 BoardConfig.mk文件的作用

## 2.3.5 AndroidProducts.mk文件

## 2.3.6 AndroidBoard.mk文件

## 2.3.7 编译中的层叠加

## 第3章 本地层基础部分

## 3.1 本地层基础部分结构

## 3.2 C语言基础：Bionic

## 3.2.1 Bionic各个部分的结构

## 3.2.2 Bionic的头文件

## 3.2.3 内存泄露的检查库

## 3.2.4 属性系统的支持

## 3.3 C语言底层库

## 3.3.1 Log库

## 3.3.2 C语言工具库libcutils

## 3.3.3 属性和套接字支持

## 3.4 init可执行程序和相关脚本

## 3.4.1 init可执行程序的结构

## 3.4.2 启动脚本init.rc

## 3.4.3 init的内建动作和初始化

- 3.4.4 init的循环
- 3.4.5 设备管理功能
- 3.5 shell工具
  - 3.5.1 shell程序sh
  - 3.5.2 命令工具箱toolbox
- 3.6 Android属性和套接字机制
  - 3.6.1 属性机制
  - 3.6.2 套接字机制
- 第4章 本地层框架部分
  - 4.1 本地层框架部分结构
  - 4.2 框架层的工具库libutils
    - 4.2.1 libutils的结构
    - 4.2.2 基本的定义和类型
    - 4.2.3 数据结构和容器类
    - 4.2.4 工具类
    - 4.2.5 引用计数和强弱引用指针
    - 4.2.6 线程类及相关辅助功能
    - 4.2.7 文件操作的监听循环
  - 4.3 作为全局守护进程的可执行程序
    - 4.3.1 app\_process
    - 4.3.2 keystore
    - 4.3.3 installd
    - 4.3.4 vold
    - 4.3.5 netd
  - 4.4 Binder的IPC机制结构
    - 4.4.1 Android Binder机制的层次结构
    - 4.4.2 具体Binder的实现
  - 4.5 Binder库以及相关内容
    - 4.5.1 具体Binder的实现
    - 4.5.2 Binder库的核心功能
    - 4.5.3 Binder框架和具体IPC结构的实现
    - 4.5.4 IServiceManager相关
    - 4.5.5 Binder库中的其他部分
  - 4.6 servicemanager可执行程序和服务的实现
    - 4.6.1 服务管理器可执行程序和服务的实现
    - 4.6.2 在系统中实现“服务”
    - 4.6.3 servicemanager对服务的管理
    - 4.6.4 Android中本地服务的层次结构
- 第5章 本地的启动流程、调试和设计结构
  - 5.1 Android启动流程的本地阶段

- 5.1.1 启动流程概述
- 5.1.2 BootLoader运行及加载  
内核
- 5.1.3 Linux内核运行和加载根文件  
系统
- 5.1.4 运行用户空间的第一个  
进程
- 5.1.5 挂接各个文件系统
- 5.1.6 系统各个进程的运行
- 5.1.7 启动图
- 5.2 Android本地部分的调试
- 5.2.1 标准Linux调试方法
- 5.2.2 使用log的方法
- 5.2.3 Android的特殊调试工具
- 5.3 本地程序的结构设计
- 5.3.1 目标和设计思路
- 5.3.2 守护进程+保留Socket的  
设计结构
- 5.3.3 守护进程+本地层Binder的  
设计结构
- 第6章 Java运行环境
- 6.1 Dalvik虚拟机和相关工具
- 6.1.1 Dalvik的概述
- 6.1.2 虚拟机的实现和运行
- 6.1.3 Dalvik工具
- 6.1.4 nativehelper库
- 6.2 Java基础库libcore
- 6.2.1 libcore的源代码结构
- 6.2.2 luni目录中的内容
- 6.2.3 libcore中的其他内容
- 第7章 Java框架层
- 7.1 Java框架层结构
- 7.2 Android的Java层的API
- 7.2.1 Java框架层API的含义
- 7.2.2 API的描述文件
- 7.2.3 被隐藏的内容
- 7.2.4 不赞成使用的（过时的）API
- 7.3 Java框架库framework.jar
- 7.3.1 框架库的组成和作用
- 7.3.2 框架库的API
- 7.3.3 框架库的编译结构
- 7.3.4 框架库的各个部分
- 7.3.5 属性机制
- 7.3.6 套接字机制
- 7.4 Java服务库services.jar
- 7.4.1 服务库的组成和作用
- 7.4.2 服务管理器

### 7.4.3 主要的服务功能

### 7.4.4 启动结束的处理

## 7.5 资源包framework-res.apk

### 7.5.1 资源包的组成和作用

### 7.5.2 作为API的资源id

### 7.5.3 资源包AndroidManifest.Xml

## 文件

## 7.6 策略库

### 7.6.1 策略库的组成和作用

### 7.6.2 策略库的功能和使用

## 7.7 Java层的Binder机制

### 7.7.1 Java Binder的框架

### 7.7.2 Java框架层的Binder使用

### 7.7.3 Binder中参数和返回值的类型

## 第8章 Java本地调用

### 8.1 Android的Java本地调用

#### 8.1.1 JNI的基本概念

#### 8.1.2 Android的JNI实现基础

#### 8.1.3 Android中JNI的存在方式

### 8.2 在框架中使用JNI

#### 8.2.1 框架层使用JNI的结构

#### 8.2.2 框架层JNI的实现

### 8.3 在应用程序包中使用JNI

#### 8.3.1 应用程序包的JNI部分

#### 8.3.2 应用程序包中JNI的典型实现方式

#### 8.3.3 结构上的几种变化

### 8.4 JNI的高级使用

#### 8.4.1 JNI的双向调用

#### 8.4.2 线程的处理

#### 8.4.3 JNIHelp中的特殊处理

## 第9章 Java启动流程、调试和程序结构

### 9.1 Java部分的启动流程

#### 9.1.1 启动流程概述

#### 9.1.2 Java系统的守护进程

#### 9.1.3 ZygoteInit 类的执行

#### 9.1.4 系统服务器部分

#### 9.1.5 Java应用程序部分的启动

### 9.2 Java部分的调试

#### 9.2.1 在Java中使用Log

#### 9.2.2 Java层的命令行程序

#### 9.2.3 代码内容相关的调试

#### 9.2.4 DDMS工具

#### 9.2.5 HierarchyViewer工具

### 9.3 Java层程序的结构设计

- 9.3.1 目标和设计思路
- 9.3.2 在系统服务器中服务的设计结构
- 9.3.3 Service组件的设计结构
- 第10章 GUI系统和应用程序管理
- 10.1 GUI系统的结构和模块关系
- 10.2 UI框架库及相关
  - 10.2.1 基本类型方面
  - 10.2.2 显示相关
  - 10.2.3 按键和事件的处理
  - 10.2.4 其他内容
- 10.3 Surface图层系统
  - 10.3.1 Surface系统的结构
  - 10.3.2 Surface本地的接口部分
  - 10.3.3 Surface本地实现部分
  - 10.3.4 Surface的JNI代码
  - 10.3.5 Surface的Java部分
- 10.4 2D图形系统
  - 10.4.1 2D图形系统的结构
  - 10.4.2 Skia底层库
  - 10.4.3 图形系统的JNI接口
  - 10.4.4 2D图形包
- 10.5 3D图形系统
  - 10.5.1 OpenGL系统的结构
  - 10.5.2 OpenGL的本地代码
  - 10.5.3 OpenGL的JNI代码
  - 10.5.4 OpenGL的Java类
- 10.6 窗口和视图部分
  - 10.6.1 窗口和视图部分的结构
  - 10.6.2 窗口管理的底层实现
  - 10.6.3 视图实现和上层的窗口管理器的实现
- 10.7 活动和应用程序管理
  - 10.7.1 活动和应用程序管理的结构
  - 10.7.2 活动界面的实现
  - 10.7.3 应用程序管理的实现
- 第2篇 Android系统级应用
- 开发要点
- 第11章 Android应用层接口结构与安全
  - 11.1 Android应用层接口结构
    - 11.1.1 标准应用程序和源代码开发的应用程序
    - 11.1.2 第三方应用程序和预置应用程序
  - 11.2 Android的安全系统



- 11.2.1 安全机制概述
- 11.2.2 Linux的用户和组
- 11.2.3 Android的许可
- 11.3 程序签名问题
  - 11.3.1 Android应用签名的概念
  - 11.3.2 Android应用签名的流程
  - 11.3.3 不同的程序签名
  - 11.3.4 证书生成工具
- 11.4 安全策略
  - 11.4.1 安全策略的几个方面
  - 11.4.2 安全策略的实现
  - 11.4.3 安全策略的定制
- 第12章 应用层的本地内容和NDK开发
  - 12.1 本地应用的标准化
  - 12.2 NDK环境
    - 12.2.1 NDK的基本概念
    - 12.2.2 从几个方面理解NDK的不同含义
  - 12.3 NDK的组成结构
    - 12.3.1 NDK环境的结构
    - 12.3.2 NDK中包含的内容
  - 12.4 NDK开发基础
    - 12.4.1 编译流程
    - 12.4.2 NDK编程要点
    - 12.4.3 开发结构
  - 12.5 使用JNI的方式
    - 12.5.1 NDK中使用JNI工程结构
    - 12.5.2 在本地操作Bitmap位图
    - 12.5.3 在本地使用OpenGL ES绘制
    - 12.5.4 程序实现
  - 12.6 在本地构建应用
    - 12.6.1 本地应用接口基础
    - 12.6.2 实现本地活动的辅助库：  
android\_native\_app\_glue
    - 12.6.3 程序实现
- 第13章 预置的应用程序层
  - 13.1 预置应用层程序概述
  - 13.2 预置综合应用程序
    - 13.2.1 主屏幕
    - 13.2.2 设置
    - 13.2.3 联系人和电话
    - 13.2.4 短信息和彩信
    - 13.2.5 浏览器和邮件
    - 13.2.6 音乐

- 13.2.7 相机和录音机
- 13.2.8 画廊
- 13.2.9 搜索
- 13.2.10 时钟、计算器和日历
- 13.3 预置内容提供者
  - 13.3.1 联系人信息提供者
  - 13.3.2 媒体信息提供者
  - 13.3.3 电话信息提供者
  - 13.3.4 下载信息提供者
- 13.4 预置输入法
- 13.5 活动墙纸系统
  - 13.5.1 活动墙纸选择器
  - 13.5.2 几个具体的活动墙纸
- 13.6 实现框架功能的应用程序包
  - 13.6.1 系统界面
  - 13.6.2 设置信息提供者
  - 13.6.3 几个主体为Service组件的应用程序包
- 第14章 Android系统定制
  - 14.1 Android系统定制概述
  - 14.2 普通应用程序的定制
    - 14.2.1 普通应用定制的方面
    - 14.2.2 预置的应用程序的组件可见性
    - 14.2.3 组件的intent-filter
  - 14.3 桌面的定制
    - 14.3.1 桌面程序的入口和功能
    - 14.3.2 桌面程序细节
    - 14.3.3 在桌面之前执行的程序
  - 14.4 系统资源的定制
    - 14.4.1 定义资源的概念和原则
    - 14.4.2 一些配置的数值
    - 14.4.3 可绘制内容的定制
    - 14.4.4 样式和主题的定制
    - 14.4.5 内部布局的定制
  - 14.5 框架内部组件和界面的定制
    - 14.5.1 内部组件
    - 14.5.2 内部界面
  - 14.6 状态栏界面的定制
    - 14.6.1 状态栏界面的功能和布局
    - 14.6.2 状态栏界面
  - 14.7 锁屏界面的定制
    - 14.7.1 锁屏界面的功能
    - 14.7.2 锁屏界面
  - 14.8 开机动画的定制
- 第15章 系统级的调试和测试

## 15.1 调试

### 15.1.1 命令行调试工具

### 15.1.2 界面上的辅助工具

## 15.2 测试系统和工具

### 15.2.1 测试系统的结构

### 15.2.2 Monkey

### 15.2.3 MonkeyRunner

## 15.3 代码级别的测试

### 15.3.1 代码测试框架

### 15.3.2 代码测试方式

## 15.4 兼容性及兼容性测试

### 15.4.1 Android兼容性

### 15.4.2 CTS兼容性验证

### 15.4.3 CTS的相关源代码和原理

## 附录A Android版本演进和

### Android 4.x

#### A.1 Android版本的演进

#### A.2 编译系统的更新

#### A.3 本地框架层的更新

##### A.3.1 init.rc中的关键字

##### A.3.2 surfaceflinger作为独立的守护进程

##### A.3.3 netd守护进程

#### A.4 Java框架的更新

##### A.4.1 Java框架的API描述

##### A.4.2 Java类库的升级

##### A.4.3 Java服务包

##### A.4.4 工具

#### A.5 GUI系统的更新

##### A.5.1 GUI系统

##### A.5.2 Skia引擎的改动

##### A.5.3 OpenGL引擎的改动

##### A.5.4 窗口的变化

#### A.6 Java应用程序层的更新

##### A.6.1 SystemUI

##### A.6.2 桌面

##### A.6.3 普通应用程序

#### A.7 本地标准应用和NDK环境

## 章节摘录

版权页：插图：1.活动管理器服务（ActivityManagerService）活动管理服务定义了系统组件层次的管理功能，是服务库各个功能模块中比较复杂的一个。

am子目录主要负责活动管理功能，其中的主文件是ActivityManagerService.java。

活动管理器服务的主要ActivityManagerService类继承了ActivityManagerNative，实际上继承了以下文件中的接口：frameworks/base/core/java/android/app/IActivityManager.java。

活动管理器服务提供的主要功能如下所示。

- 各个组件的注册和许可（包括Uri许可）的检查。

- Activity组件的启动、返回结果的处理，使用观测者（IActivityWatcher）。

- Service组件的启动、停止、绑定、解除绑定。

- BoardcastReceiver组件的注册、注销，广播的发送。

- 处理应用程序的状态（进程信息的设置获取、杀死进程、应用的错误、崩溃）。

- 关机、唤醒和睡眠等操作。

- 安装系统级别的ContentProvider。

相比其他服务，ActivityManagerService偏重于更上层、综合性的内容。

其管是表示应用程序的Application（包括其运行的进程上下文），Activity、Service、BoardcastReceiver和ContentProvider4种组件，以及相关的Permission。

2.窗口管理器服务（WindowManagerService）窗口管理器服务的主文件是WindowManagerService.java，负责应用程序中窗口相关内容的处理。

WindowManagerService的实现Android GUI系统的运行密切相关。

WindowManagerService类实现了以下文件中定义的接口：

frameworks/base/core/java/android/view/IWindowManager.aidl

frameworks/base/core/java/android/view/IWindowSession.aidl WindowManagerService提供的主要功能如下所示。

- 窗口和View的构建和行为的处理。

- 输入事件的处理。

- 窗口旋转、动画等处理。

- 锁屏界面等特殊屏幕的处理。

3.包管理器服务（PackageManagerService）包管理器服务的主文件是PackageManagerService.java，主要负责系统apk包安装和卸载的相关处理。

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:<http://www.tushu007.com>