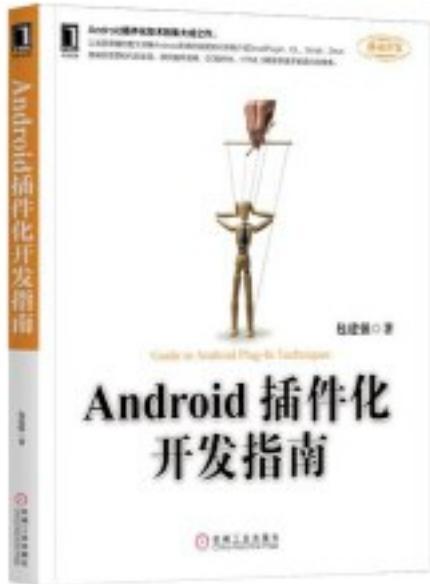


Android插件化开发指南



[Android插件化开发指南_下载链接1](#)

著者:包建强

出版者:机械工业出版社丛

出版时间:2018-8-6

装帧:平装

isbn:9787111603368

Android插件化技术广泛应用于移动开发中，通过插件化技术，AndroidApp可以不通过发布新版本而修复线上的bug或者推出新功能，这对于日活千万的App而言是非常重要的，可极大程度提升App开发效率和质量。本书详细介绍了插件化技术的原理、各种方案，及其在AndroidApp中的使用。在介绍插件化技术的同时，还详细介绍了Android底层知识，能帮助AndroidApp开发人员更好地掌握Android的开发技术。本书共22章，先介绍Android插件化技术的历史和现状，系统介绍了Android的底层知识以及反射、代理等技术；然后介绍插件化技术的基本知识，包括各类插件化解决方案及其对比；之后分析了一些插件化框架，如that、Zeus等，深入分析了资源及其在插件化中的应用，以及混淆、增量更新等周边技术的插件化解决方案。

作者介绍:

目录: 目录 Contents

序一

序二

序三

前言

一部分 预备知识

1章 插件化技术的昨天、今天与明天2

1.1 插件化技术是什么2

1.2 为什么需要插件化3

1.3 插件化技术的历史3

1.4 插件化技术的用途到底是什么8

1.5 更好的替代品: React Native8

1.6 只有中国这么玩吗9

1.7 四大组件都需要插件化技术吗10

1.8 双开和虚拟机10

1.9 从原生页面到HTML 5的过渡11

1.10 本章小结12

2章 Android底层知识13

2.1 概述13

2.2 Binder原理14

2.3 AIDL原理16

2.4 AMS20

2.5 Activity工作原理21

2.5.1 App是怎么启动的21

2.5.2 启动App并非那么简单21

2.6 App内部的页面跳转32

2.7 Context家族史34

2.8 Service工作原理36

2.8.1 在新进程启动Service36

2.8.2 启动同一进程的Service39

2.8.3 在同一进程绑定Service39

2.9 BroadcastReceiver工作原理41

2.9.1 注册过程43

2.9.2 发送广播的流程44

2.9.3 广播的种类45

2.10 ContentProvider工作原理46

2.10.1 ContentProvider的本质49

2.10.2 匿名共享内存 (ASM) 49

2.10.3 ContentProvider与AMS的通信流程50

2.11 PMS及App安装过程52

2.11.1 PMS简介52

2.11.2 App的安装流程52

2.11.3 PackageParser53

2.11.4 ActivityThread与PackageManager54

2.12 ClassLoader家族史55

2.13 双亲委托57

2.14 MultiDex57

2.15 实现一个音乐播放器App59

2.15.1 基于两个Receiver的音乐播放器59

2.15.2 基于一个Receiver的音乐播放器63

2.16 本章小结68

3章 反射70

3.1 基本反射技术	70
3.1.1 根据一个字符串得到一个类	70
3.1.2 获取类的成员	71
3.1.3 对泛型类的反射	75
3.2 jOOR	77
3.2.1 根据一个字符串得到一个类	78
3.2.2 获取类的成员	78
3.2.3 对泛型类的反射	79
3.3 对基本反射语法的封装	80
3.3.1 反射出一个构造函数	81
3.3.2 调用实例方法	81
3.3.3 调用静态方法	82
3.3.4 获取并设置一个字段的值	82
3.3.5 对泛型类的处理	83
3.4 对反射的进一步封装	84
3.5 本章小结	88
4章 代理模式	89
4.1 概述	89
4.1.1 远程代理 (AIDL)	90
4.1.2 保护代理 (权限控制)	92
4.1.3 虚代理 (图片占位)	92
4.1.4 协作开发 (Mock Class)	92
4.1.5 给生活加点料 (记日志)	93
4.2 静态代理和动态代理	94
4.3 对AMN的Hook	95
4.4 对PMS的Hook	97
4.5 本章小结	98
5章 对startActivity方法进行Hook	99
5.1 startActivityForResult方法的两种形式	99
5.2 对Activity的startActivity方法进行Hook	100
5.2.1 方案1：重写Activity的startActivityForResult方法	102
5.2.2 方案2：对Activity的mInstrumentation字段进行Hook	102
5.2.3 方案3：对AMN的getDefalut方法进行Hook	104
5.2.4 方案4：对H类的mCallback字段进行Hook	107
5.2.5 方案5：再次对Instrumentation字段进行Hook	109
5.3 对Context的startActivity方法进行Hook	111
5.3.1 方案6：对ActivityThread的mInstrumentation字段进行Hook	111
5.3.2 对AMN的getDefalut方法进行Hook是一劳永逸的	113
5.4 启动没有在AndroidManifest中声明的Activity	113
5.4.1 “欺骗AMS”的策略分析	114
5.4.2 Hook的上半场	115
5.4.3 Hook的下半场：对H类的mCallback字段进行Hook	118
5.4.4 Hook的下半场：对ActivityThread的mInstrumentation字段进行Hook	119
5.4.5 “欺骗AMS”的弊端	121
5.5 本章小结	121
二部分 解决方案	
6章 插件化技术基础知识	124
6.1 加载外部的dex	124
6.2 面向接口编程	126
6.3 插件的瘦身	129
6.4 对插件进行代码调试	131
6.5 Application的插件化解决方案	133
6.6 本章小结	134
7章 资源初探	135

7.1 资源加载机制	135
7.1.1 资源分类	135
7.1.2 剪不断理还乱：Resources和AssetManager	136
7.2 资源的插件化解决方案	137
7.3 换肤	141
7.4 殊途同归：另一种换肤方式	149
7.5 本章小结	149
8章 简单的插件化解决方案	150
8.1 在AndroidManifest中声明插件中的组件	150
8.2 宿主App加载插件中的类	151
8.3 启动插件Service	152
8.4 加载插件中的资源	152
8.5 本章小结	154
9章 Activity的插件化解决方案	155
9.1 启动没有在AndroidManifest中声明的插件Activity	155
9.2 基于动态替换的Activity插件化解决方案	159
9.2.1 Android启动Activity的原理分析	159
9.2.2 故意命中缓存	160
9.2.3 加载插件中类的方案1：为每个插件创建一个ClassLoader	164
9.2.4 为了圆一个谎言而编造更多的谎言	164
9.3 加载插件中类的方案2：合并多个dex	166
9.4 为Activity解决资源问题	169
9.5 对LaunchMode的支持	169
9.6 加载插件中类的方案3：修改App原生的ClassLoader	172
9.7 本章小结	174
10章 Service的插件化解决方案	175
10.1 Android界的荀彧和荀攸：Service和Activity	175
10.2 预先占位	176
10.3 startService的解决方案	178
10.4 bindService的解决方案	183
10.5 本章小结	185
11章 BroadcastReceiver的插件化解决方案	186
11.1 Receiver概述	186
11.2 动态广播的插件化解决方案	187
11.3 静态广播的插件化解决方案	187
11.4 静态广播的插件化解决方案	189
11.5 本章小结	193
12章 ContentProvider的插件化解决方案	194
12.1 ContentProvider基本概念	194
12.2 一个简单的ContentProvider例子	195
12.3 ContentProvider插件化	197
12.4 执行这段Hook代码的时机	199
12.5 ContentProvider的转发机制	200
12.6 本章小结	201
13章 基于静态代理的插件化解决方案：that框架	202
13.1 静态代理的思想	202
13.2 一个简单的静态代理的例子	203
13.2.1 从宿主Activity跳转到插件Activity	203
13.2.2 ProxyActivity与插件Activity的通信	204
13.2.3 插件Activity的逻辑	206
13.3 插件内部的页面跳转	206
13.4 从“肉体”上消灭that关键字	207
13.5 插件向外跳转	209
13.6 面向接口编程在静态代理中的应用	211

13.7 对LaunchMode的支持	216
13.8 本章小结	221
14章 that框架对Service和BroadcastReceiver的支持	222
14.1 静态代理的思想在Service的应用	222
14.2 对BindService的支持	227
14.3 Service的预先占位思想	229
14.4 Service插件化的解决方案：动静结合	231
14.4.1 解析插件中的Service	231
14.4.2 通过反射创建一个Service对象	232
14.4.3 ProxyService与ServiceManager	234
14.4.4 bindService的插件化解决方案	240
14.5 静态代理的思想在BroadcastReceiver的应用	245
14.6 本章小结	248
15章 再谈资源	249
15.1 Android App的打包流程	249
15.2 修改AAPT	251
15.2.1 修改并生成新的aapt命令	251
15.2.2 在插件化项目中使用新的aapt命令	254
15.3 public.xml固定资源id值	256
15.4 插件使用宿主的资源	258
15.5 本章小结	259
16章 基于Fragment的插件化框架	261
16.1 AndroidDynamicLoader概述	261
16.2 简单的Fragment插件化例子	262
16.3 插件内部的Fragment跳转	263
16.4 从插件的Fragment跳转到插件外部的Fragment	264
16.5 本章小结	266
三部分 相关技术	
17章 降级	268
17.1 从Activity到HTML	5269
17.2 从HTML 5到Activity	273
17.3 对返回键的支持	278
17.4 本章小结	278
18章 插件的混淆	279
18.1 插件的基本混淆	279
18.2 方案1：不混淆公共库MyPluginLibrary	280
18.3 方案2：混淆公共库MyPluginLibrary	282
18.3.1 配置multidex	283
18.3.2 配置proguard	285
18.3.3 移除Plugin1.apk中的冗余dex	286
18.4 本章小结	287
19章 增量更新	288
19.1 如何使用增量更新	288
19.2 制作插件的增量包	289
19.3 App下载增量包并解压到本地	289
19.4 App合并增量包	290
19.5 本章小结	291
20章 so的插件化解决方案	292
20.1 编写一个简单的so	292
20.2 使用so	296
20.3 so的加载原理	298
20.4 基于System.load的插件化解决方案	301
20.5 基于System.loadLibrary的插件化解决方案	304
20.6 本章小结	305

21章 对App的打包流程进行Hook	306
21.1 自定义Gradle插件	306
21.1.1 创建Gradle自定义插件项目	306
21.1.2 Extension又是什么	308
21.1.3 修改打包流程	309
21.2 修改resources.arsc	311
21.2.1 Android是怎么查找资源的	311
21.2.2 aapt都干了什么	312
21.2.3 gradle-small的工作原理	313
21.2.4 怎么使用gradle-small	313
21.2.5 gradle-small中的Plugin家族	313
21.2.6 gradle-small中的Editor家族	317
21.3 本章小结	318
22章 插件化技术总结	319
22.1 插件的工程化	319
22.2 插件中类的加载	319
22.3 哪些地方可以Hook	320
22.4 Activity插件化的解决方案	320
22.5 资源的解决方案	321
22.6 Fragment是哪门哪派	322
22.7 Service、ContentProvider、BroadcastReceiver插件化的通用解决方案	322
22.8 特定于Service的插件化解决方案	322
22.9 特定于BroadcastReceiver的插件化解决方案	323
22.10 特定于ContentProvider的插件化解决方案	323
22.11 本章小结	323
附录	
附录A 常用工具	326
附录B 本书代码索引	328
· · · · · (收起)	

[Android插件化开发指南_下载链接1](#)

标签

Android

插件化

Android进阶

互联网

2019

计算机

编程

Android高级进阶

评论

源码处再分析得细点就好了

水平跟阿里的热更新那本差很多，可能是面向新手吧。如果是把核心思路抽成几章长文感觉会更好。

代码写得随意了点

书不错，书中详细介绍了几种插件的原理，对于了解插件化有帮助。

回顾和总结了这些年Android插件化涉及到的知识点，浅显易懂。

对插件化的基础理论做了详细的讲述

四个字概括：泛泛而谈！

本书包含的内容很多，但是每个知识点几乎都是泛泛而谈，没有一个讲出所以然的，恕我直言，感觉作者功力不够，或者有“到处凑内容”之嫌，有些章节看标题挺好，但内容要么不对题，要么草草收场，真不如看一些质量挺高的博客。

挺好，对插件化理解很深的书

插件化入门书籍

[Android插件化开发指南_下载链接1](#)

书评

序 DroidPlugin

DroidPlugin使用了一些比较hack的技巧，但是总结起来也就是一句话“利用hook技术实现欺上瞒下，从而达到免安装运行的目的”。因为Android系统出于安全考虑，系统服务与App进程采用分进程设计，它们之间通讯使用binder技术，系统服务实际上不知道App进程中运行...

[Android插件化开发指南_下载链接1](#)