



(12) 发明专利申请

(10) 申请公布号 CN 103399792 A

(43) 申请公布日 2013. 11. 20

(21) 申请号 201310317921. 6

(22) 申请日 2013. 07. 24

(71) 申请人 佳都新太科技股份有限公司
地址 510665 广东省广州市天河软件园建工路 4 号

(72) 发明人 刘瑞斌 唐敏 宁学军

(51) Int. Cl.
G06F 9/48 (2006. 01)

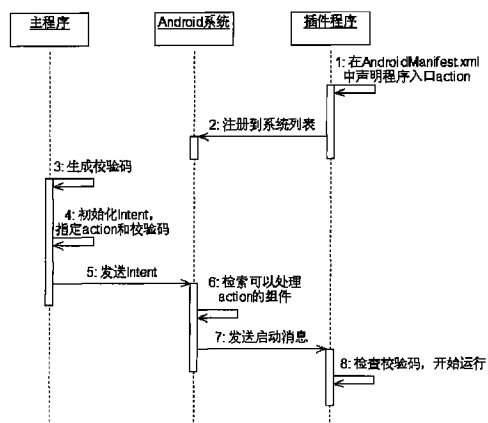
权利要求书1页 说明书3页 附图2页

(54) 发明名称

基于 Android 的应用插件化实现方案

(57) 摘要

本发明公开了基于 Android 的应用插件化实现方案,主要应用于开发插件化形式的 Android 应用程序,解决了应用程序体积大、功能复杂、无法部分更新的问题,本发明特点在于,Android 应用插件化,将非核心功能与主程序分离,减小主程序的体积和资源消耗;可动态配置 Android 应用程序的插件,包括配置插件和插件版本;插件用于扩展 Android 应用的功能,通过服务器提供的插件列表,动态更新插件;插件版本,用于对比插件的当前版本和最新版本,当发现插件的当前版本旧于最新版本时,Android 应用可选择更新插件。



1. 基于 Android 的应用插件化实现方案,其特征在於,包括:

将应用整体拆分成主体应用部分和插件应用部分,主体部分和插件部分均为 APK(Android Package) 安装程序,均可独立安装,但主体部分可独立启动,插件部分只能通过主体部分启动。

2. 根据权利要求 1 所述的方法,其特征在於,还包括:主体部分和插件部分通过 Android 的隐式 Intent 机制桥接,每个插件入口对应一个 action 字段,主体部分通过指定 Intent 的 action,来启动插件程序。

3. 根据权利要求 1 所述的方法,其特征在於,还包括:

通过 Android 的 permission 机制,加强插件部分的使用安全,插件部分需预先声明 permission 要求,主体部分只有声明了改 permission,才允许启动插件程序;

通过增加插件启动参数,在插件程序的入口部分增加参数校验代码,对主体部分传入的参数进行校验,如果校验成功,则允许使用启动插件,否则,不允许启动插件。

4. 基于 Android 的应用插件化实现方案,其特征在於,包括服务端程序和客户端程序;

所述服务端程序,用于:配置客户端的插件,当客户端向服务端请求插件列表时,将插件列表通过网络返回给客户端,客户端通过该插件列表,可选择性地安装插件;

所述客户端程序,用于:获取插件列表后,提供插件列表显示,插件下载,插件安装,插件启动,插件卸载等功能。

基于 Android 的应用插件化实现方案

技术领域

[0001] 本发明涉及移动互联网领域,特别是涉及基于 Android 的应用插件化实现方案。

背景技术

[0002] 随着移动互联网迅猛发展,智能手机、平板电脑已能够完成大部分 PC 的所实现的功能。由于这些移动设备具有轻便,美观,易用的特点,深得大众的喜爱。然而,移动设备也受限于电量,体积等因素,在 CPU、内存和电池等资源比较短缺。电池电量资源在 Android 系统中显得尤其短缺。电池的消耗主要来源于频繁的网络访问。

[0003] Android 系统是一个允许多任务的操作系统,用户退出所有 Activity 后,线程、服务等都允许在继续后台运行,通过这种机制,应用可以维护自身的正确性和可用性。当下,优质的应用程序使用更智能的机制,来管理程序的线程、服务等,避免不必要的网络访问,采用预加载机制,监听手机充电状态等,来减少程序消耗的资源。尽管如此,假如这些功能并非用户的实际需求,便对资源造成了不必要的浪费。

发明内容

[0004] 本发明所要解决的技术问题是提供基于 Android 的应用插件化实现方案,以节省现有技术中 Android 应用程序对电量、CPU、内存等资源的消耗。

[0005] 为实现上述发明目的,本发明提供基于 Android 的应用插件化实现方案,包括 Android 应用程序和服务端应用程序,所述 Android 应用程序分为一个 Android 主应用程序和若干个 Android 插件应用程序;

[0006] 所述 Android 主应用程序,用于:管理 Android 插件应用程序,通过与服务器交互,维护可用插件列表,同时允许启动插件应用程序;

[0007] 所述 Android 插件应用程序,用于:具体实现应用程序的特定功能模块;

[0008] 所述服务端应用程序,用于:配置应用程序的插件列表,提供插件的下载功能。

[0009] 本发明还提供基于 Android 的应用插件化实现方案,包括:

[0010] Android 主应用程序通过 Intent 启动 Android 插件应用程序,Intent 的 action 字段为识别启动指定插件应用程序的唯一标识;

[0011] Android 插件应用程序在其配置文件中声明被主程序调用的入口,可以为指定 action 的 Activity 或 Service 或 BroadcastReceiver,插件应用程序没有主动的入口,只能通过主应用程序启动。

[0012] 本发明还提供基于 Android 的应用插件化实现方案,包括:

[0013] Android 主应用程序启动 Android 插件应用程序时,传入校验码,插件应用程序检验校验码,如果校验码通过检查,则允许正常使用插件应用程序,否则,给予适当提示,并退出插件应用程序。

附图说明

[0014] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单的介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0015] 图 1 为本发明实施例一中流程示意图;

[0016] 图 2 为本发明实施例二中流程示意图。

具体实施方式

[0017] 为使本发明的上述目的、特征和优点能够更加明显易懂,下面结合附图和具体实施方式对本发明作进一步详细的说明。显然,所描述的实施例仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0018] 实施例一,参见图 1。

[0019] 本发明提供基于 Android 的应用插件化实现方案,所述方案包括 Android 主应用程序和 Android 插件应用程序,其中:

[0020] 所述 Android 主应用程序,用于生成校验码和启动插件程序;

[0021] 所述 Android 插件应用程序,用于注册到系统列表,允许主程序调用,并对启动校验码进行校验,确保应用安全。

[0022] 上述方案的工作过程,具体包括以下步骤:

[0023] 步骤 1,插件程序在 AndroidManifest.xml 中声明程序的入口 action,入口组件可以是 Activity 或 Service 或 Broadcast Receiver;

[0024] 步骤 2,将应用程序打包成 apk 安装文件,安装到 Android 设备,插件程序自动注册到系统列表;

[0025] 步骤 3,主程序根据预先定制的规则,生成启动插件的校验码;

[0026] 步骤 4,主程序初始化 Intent 对象,指定 action 为步骤 1 中定义的 action,并将步骤 3 中生成的校验码作为参数添加到 Intent 对象中;

[0027] 步骤 5,主程序通过 startActivity 或 startService 或 sendBroadcast 方法发送 Intent 给 Android 系统;

[0028] 步骤 6,Android 系统检索已经注册的组件,得出 Intent 的 action 与在系统列表注册的 action 相匹配的组件;

[0029] 步骤 7,Android 系统向步骤 6 中匹配的组件发送启动消息;

[0030] 步骤 8,插件程序收到启动消息,检查 Intent 的校验码,如果校验码正确,则启动程序,否则,不允许启动程序。

[0031] 实施例二,参见图 2。

[0032] 基于实施例一中基于 Android 的应用插件化实现方案,还包括:Android 主程序与服务器程序交互,管理插件应用程序,其中:

[0033] 步骤 1,Android 主程序向服务器程序发起获取插件列表的请求,等待服务器返回;

[0034] 步骤 2,服务器程序根据请求,查询符合该程序的应用列表;

[0035] 步骤 3,服务器将插件列表返回给 Android 主程序;

[0036] 步骤 4,Android 主程序根据插件列表,向服务器请求下载插件程序;

[0037] 步骤 5,服务器将插件安装程序返回给 Android 主程序;

[0038] 步骤 6,当用户不需要插件时,主程序将插件程序卸载。

[0039] 由上可见,本发明提供的基于 Android 的应用插件化实现方案,有以下优点。

[0040] (1) 实现简单,便于维护

[0041] 本发明只需主程序与插件程序之间的协议,将 Intent 的 action 和校验码融合在一个私有协议,即可实现应用插件化,无需将所有功能都集成到一个安装包中,当插件程序有更新时,可以只更新插件程序,不更新主程序,插件程序之间互不影响。

[0042] (2) 安全性高

[0043] 首先,插件程序需在 AndroidManifest.xml 配置文件中声明 permission 自定义权限,如不知道权限内容,则无法启动插件程序;其次,插件程序的启动入口处对校验码进行检验,避免恶意启动插件程序,对程序的数据造成破坏。

[0044] (3) 节省资源,提高应用质量

[0045] 通过分离核心功能和非核心功能,将非核心功能独立为插件程序,仅在需要时加载插件程序,假如没有安装插件程序,则表明非核心功能完全不消耗设备的资源,良好的资源管理,有助于提升应用的质量。

[0046] 以上所述仅是本发明的具体实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

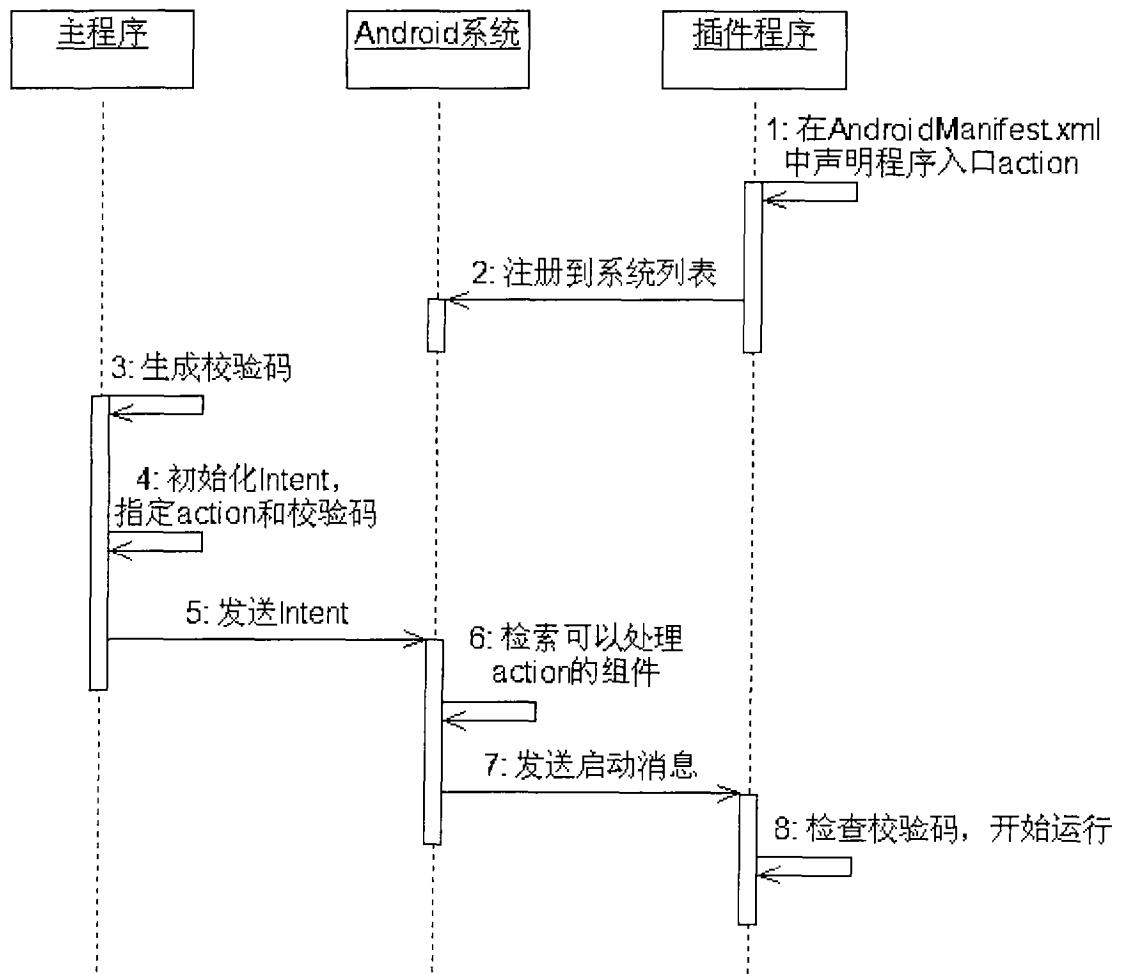


图 1

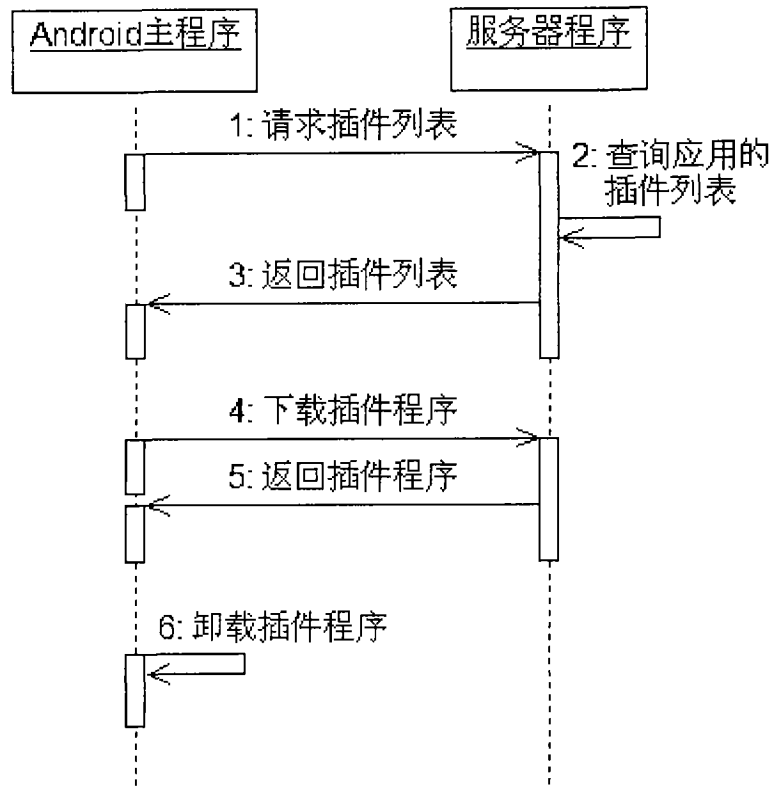


图 2