



(12) 发明专利

(10) 授权公告号 CN 102622241 B

(45) 授权公告日 2016.06.15

(21) 申请号 201110028778.X

(22) 申请日 2011.01.26

(73) 专利权人 腾讯科技(深圳)有限公司

地址 518044 广东省深圳市福田区振兴路赛格科技园2栋东403室

(72) 发明人 马斌

(74) 专利代理机构 北京派特恩知识产权代理有限公司 11270

代理人 程立民 张颖玲

(51) Int. Cl.

G06F 9/445(2006.01)

(56) 对比文件

CN 101110788 A, 2008.01.23, Gísli Hjálmtÿ sson. Dynamic C++ Classes A lightweight mechanism to update code in a running program. 《Proceedings of the USENIX Annual

Technical Conference》. 1998, (第98期), Gísli Hjálmtÿ sson. Dynamic C++ Classes A lightweight mechanism to update code in a running program. 《Proceedings of the USENIX Annual Technical Conference》. 1998, (第98期),

审查员 于潇

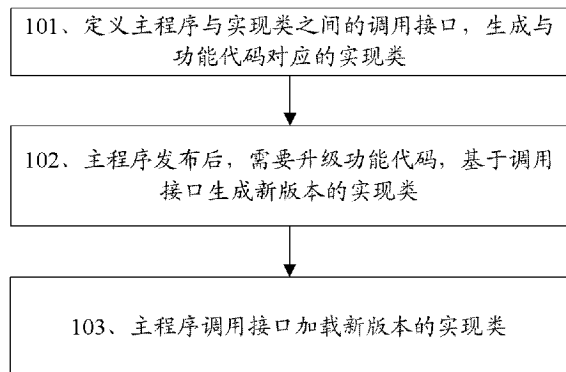
权利要求书2页 说明书6页 附图1页

(54) 发明名称

一种软件升级方法及装置

(57) 摘要

本发明公开了一种软件的升级方法及装置,用于解决移动终端的应用软件更新占用网络流量大,更新繁琐不便的问题。本发明将主程序中完成具体程序功能的功能代码提取出来,对其进行封装形成实现类,并抽象出调用接口,主程序通过所述调用接口调用所述实现类,以完成具体的程序功能;当需要对功能代码进行升级时,主程序获取新版本的实现类,用新版本的实现类替换旧版本的实现类。本发明软件在进行升级时,只需从服务器下载部分逻辑的代码文件,替换掉原有的代码逻辑,由于更新的内容只是总体程序代码的一个子集,因此占用的网络流量小。在静默升级模式下,在用户使用软件的过程中软件自动完成升级过程,从而达到静默更新的目的。



1. 一种软件升级方法,其特征在于,该方法包括:

将主程序中完成具体程序功能的功能代码提取出来,对其进行封装形成实现类,并抽象出调用接口,主程序通过所述调用接口调用所述实现类,以完成具体的程序功能;

当需要对功能代码进行升级时,主程序获取新版本的实现类,用新版本的实现类替换旧版本的实现类;

其中,在编译新版本的实现类时,将包含旧版本实现类的项目作为包含新版本实现类的项目的一个引用,并且在编译新版本实现类的项目时,通过配置编译选项过滤掉新版本实现类和旧版本实现类都使用到的会导致重名冲突的类,使得新版本的实现类文件中仅仅包含自己的逻辑代码。

2. 根据权利要求1所述的方法,其特征在于,主程序判断是否有新版本的实现类需要升级的方法为:

所述实现类存储于更新库文件中,主程序在启动时,从服务器获取版本控制信息,将其与本地的版本控制信息进行比对,当版本不一致时,从服务器下载包含新版本的实现类的更新库文件,用更新库文件替换本地保存的包含旧版本实现类的更新库文件并更新本地版本控制信息;

所述版本控制信息至少包含:当前版本号、数据包签名、更新库文件的网络地址及名称、更新库文件所包含的实现类列表、调用接口的名称、实现类的名称。

3. 根据权利要求1所述的方法,其特征在于,

在主程序中设置默认的与所述功能代码对应的实现类,用于在主程序加载更新库文件中与所述功能代码对应的新版本的实现类失败时,启用该默认的实现类。

4. 根据权利要求1所述的方法,其特征在于,

当主程序发现有新版本的实现类时,提示用户选择是否更新,或自动执行静默更新。

5. 根据权利要求1至4任一项所述的方法,其特征在于,所述方法应用于基于Android操作系统及编程环境开发的程序。

6. 一种软件升级装置,其特征在于,该装置包括:

主程序,用于通过调用接口调用实现类;

调用接口,用于调用实现类;所述调用接口由功能代码抽象后获得;

实现类,用于继承所述调用接口,实现具体的程序功能,通过对完成具体程序功能的功能代码进行封装获得;

当需要对所述功能代码升级时,主程序获取新版本的实现类,用新版本的实现类替换旧版本的实现类;

其中,在编译新版本的实现类时,将包含旧版本实现类的项目作为包含新版本实现类的项目的一个引用,并且在编译新版本实现类的项目时,通过配置编译选项过滤掉新版本实现类和旧版本实现类都使用到的会导致重名冲突的类,使得新版本的实现类文件中仅仅包含自己的逻辑代码。

7. 根据权利要求6所述的装置,其特征在于,所述主程序中包含:

版本控制模块,用于在主程序启动时,从服务器获取版本控制信息,将其与本地的版本控制信息进行比对,判断所述功能代码是否需要更新,在需要更新时还负责下载包含新版本实现类的更新库文件,用更新库文件替换本地保存的包含旧版本实现类的更新库文件并

更新本地版本控制信息；

程序加载模块,用于根据版本控制信息,通过调用接口加载与功能代码对应的实现类。

8.根据权利要求7所述的装置,其特征在于,

所述版本控制模块通过版本控制信息的比对发现所述功能代码有更新时,提示用户选择是否更新,在用户选择更新或主程序采取静默更新方式的情况下,所述版本控制模块执行更新操作;

所述程序加载模块在加载更新库文件中的实现类失败时,加载主程序中设置的默认的和所述功能代码对应的实现类。

一种软件升级方法及装置

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种软件的升级方法及装置。

背景技术

[0002] 目前,基于移动终端的操作系统的应用软件在更新时通常采用的是整体更新方式,即直接将完整的更新包放在应用软件网上商店或官方提供下载,移动终端首先卸载旧版本的应用软件,然后下载安装新版本的应用软件。这样的缺陷是,应用软件整体更新占用的流量过大,用户需要频繁手动下载和更新,为用户带来不便。

[0003] 以Android系统为例,Android系统本身提供了一套程序更新的机制,通过把程序发布到appmarket,用户从appmarket下载应用。当有新版本发布到appmarket时,用户会收到通知,让用户选择是否升级。但是一个Apk包有数兆大小,如果因为某个小变更就要用户下载完整的apk进行安装,不仅会浪费用户流量,而且如果更新过频繁会对用户形成骚扰。还有在某些特殊情况下,必须让用户更新部分程序逻辑才能正常使用(如登录逻辑,后台需要变更,前台需要重新发布版本来兼容),使用android系统提供的更新方式明显不合适。

发明内容

[0004] 有鉴于此,本发明的主要目的在于提供一种软件的升级方法及装置,用于基于解决移动终端的应用软件更新占用网络流量大,用户需要频繁手动下载和更新,为用户带来不便的技术问题。

[0005] 为达到上述目的,本发明的技术方案是这样实现的:

[0006] 一种软件升级方法,该方法包括:

[0007] 将主程序中完成具体程序功能的功能代码提取出来,对其进行封装形成实现类,并抽象出调用接口,主程序通过所述调用接口调用所述实现类,以完成具体的程序功能;

[0008] 当需要对功能代码进行升级时,主程序获取新版本的实现类,用新版本的实现类替换旧版本的实现类。

[0009] 进一步地,主程序判断是否有新版本的实现类需要升级的方法为:

[0010] 所述实现类存储于更新库文件中,主程序在启动时,从服务器获取版本控制信息,将其与本地的版本控制信息进行比对,当版本不一致时,从服务器下载包含新版本的实现类的更新库文件,用更新库文件替换本地保存的包含旧版本实现类的更新库文件并更新本地版本控制信息;

[0011] 所述版本控制信息至少包含:当前版本号、数据包签名、更新库文件的网络地址及名称、更新库文件所包含的实现类列表、调用接口的名称、实现类的名称。

[0012] 进一步地,在编译新版本的实现类时,将包含旧版本实现类的项目作为包含新版本实现类的项目的一个引用,并且在编译新版本实现类的项目时,通过配置编译选项过滤掉新版本实现类和旧版本实现类都使用到的会导致重名冲突的类,使得新版本的实现类文件中仅仅包含自己的逻辑代码。

[0013] 进一步地,在主程序中设置默认的与所述功能代码对应的实现类,用于在主程序加载更新库文件中与所述功能代码对应的新版本的实现类失败时,启用该默认的实现类。

[0014] 进一步地,当主程序发现有新版本的实现类时,提示用户选择是否更新,或自动执行静默更新。

[0015] 进一步地,所述方法应用于基于Android操作系统及编程环境开发的程序。

[0016] 基于本发明提供的软件升级方法,本发明还提出一种软件升级装置,该装置包括:

[0017] 主程序,用于通过调用接口调用实现类;

[0018] 调用接口,用于调用实现类;所述调用接口由所述功能代码抽象后获得;

[0019] 实现类,用于继承所述调用接口,实现具体的程序功能,通过对完成具体程序功能的功能代码进行封装获得;

[0020] 当需要对所述功能代码升级时,主程序获取新版本的实现类,用新版本的实现类替换旧版本的实现类。

[0021] 进一步地,所述主程序中包含:

[0022] 版本控制模块,用于在主程序启动时,从服务器获取版本控制信息,将其与本地的版本控制信息进行比对,判断所述功能代码是否需要更新,在需要更新时还负责下载包含新版本实现类的更新库文件,用更新库文件替换本地保存的包含旧版本实现类的更新库文件并更新本地版本控制信息;

[0023] 程序加载模块,用于根据版本控制信息,通过调用接口加载与功能代码对应的实现类。

[0024] 进一步地,所述版本控制模块通过版本控制信息的比对发现所述功能代码有更新时,提示用户选择是否更新,在用户选择更新或主程序采取静默更新方式的情况下,所述版本控制模块执行更新操作;

[0025] 所述程序加载模块在加载更新库文件中的实现类失败时,加载主程序中设置的默认的与所述功能代码对应的实现类。

[0026] 进一步地,在编译新版本的实现类时,将包含旧版本实现类的项目作为包含新版本实现类的项目的一个引用,并且在编译新版本实现类的项目时,通过配置编译选项过滤掉新版本实现类和旧版本实现类都使用到的会导致重名冲突的类,使得新版本的实现类文件中仅仅包含自己的逻辑代码。

[0027] 本发明中,软件在进行升级时,只需从服务器下载部分逻辑的代码文件,替换掉原有的代码逻辑,由于更新的内容只是总体程序代码的一个子集,因此占用的网络流量小。在静默升级模式下,在用户使用软件的过程中软件自动完成升级过程,从而达到静默更新的目的。

附图说明

[0028] 图1为本发明提供的软件更新方法流程示意图;

[0029] 图2为本发明提供的软件更新装置的流程图。

具体实施方式

[0030] 本发明的基本思想是,将主程序中有静默更新需求的功能代码提取出来,定义成

实现类,在主程序中通过接口调用这些实现类,从而将这些与功能代码对应的实现类组装在一起实现程序的整体功能。本发明将实现类放在库文件中,在需要更新主程序中对应的功能代码时,不需要对主程序进行整体的更新,只需要基于接口定义新的与所述功能代码相对应的实现类,将包含该新的实现类的库文件下发给客户端的主程序,主程序通过接口加载新的实现类即可完成对应功能代码的升级。通过本发明可以将更新的粒度降低到功能代码对应的类级别,最大程度上减少因更新而造成的网络带宽,尤其是对于带宽资源受限,按流量收取费用的移动终端设备来说更具有现实意义。同时,本发明的静默升级方法能够最大程度上的减少用户的操作,避免更新错误,更新过程更加智能化。

[0031] 本发明中所述的主程序指程序的入口程序或能够调用其他完成一定功能的子程序的主程序模块,通过所述主程序调用不同的接口或类来实现程序的加载运行。

[0032] 本发明中所述的功能代码是指实现某一具体功能的代码段或程序逻辑,例如可以将实现用户登录的程序逻辑从主程序中抽取出来形成一段实现用户登录的功能代码,该功能代码应可封装于一个类(本发明称为实现类)中,且该类需要继承调用接口,这样,主程序才能通过调用接口调用该实现类。只要基于调用接口编写实现类,那么就以通过替换实现类来升级所述的功能代码,从而达到软件更新的目的。

[0033] 为使本发明的目的、技术方案和优点更加清楚明白,以下举实施例并参照附图,对本发明进一步详细说明。

[0034] 图1为本发明的方法流程示意图,该方法详细步骤如下:

[0035] 步骤101、对有静默需求的功能代码进行抽象,定义调用接口,并基于调用接口编写功能代码的实现类,将实现类存储于库文件中;

[0036] 在初始发布的情况下,主程序、调用接口、存储功能代码的实现类的库文件一同发布给用户。

[0037] 步骤102、在主程序发布后,若需要对功能代码进行升级,则基于调用接口定义新的实现类,新版本的实现类也基于调用接口来编写;

[0038] 步骤103、主程序在发现有新版本的实现类时,通过调用接口加载新版本的实现类;

[0039] 优选地,可在主程序中设置默认的功能代码的实现类,用于在主程序加载本地更新库文件中的与功能代码对应的实现类失败时,使用该默认的实现类,从而避免主程序因为缺少相应的功能代码而不能运行。

[0040] 优选地,主程序在发现服务器有新的更新时,可提示用户,让用户选择是否进行更新;也可不提示用户,进行静默的升级。

[0041] 以下以Android操作系统为例,说明本发明的具体实现方式,但本发明应用并不限于具体的操作系统,只要支持面向对象及接口技术的程序开发语言即可。该实施例中,主程序的编写按照普通Android程序的写法进行,对有静默更新需求的功能代码进行抽象,定义对应的调用接口,将功能代码封装到对应的实现类中,通过调用接口来调用实现类。

[0042] 以实现用户登录的功能代码为例进行说明,通常,实现用户登录的功能代码主要包括以下几个部分:

[0043] (1)接收用户输入用户名uin、密码pass,去后台查询此用户名是否需要验证码,若需要验证码,则需要拉取验证码图片,获取用户输入的验证码vcode,若不需要验证码,则服

务器返回一个字符串,作为vcode;

[0044] (2)用uin、pass、vcode去服务器验证密码是否合法;

[0045] (3)若合法则返回cookie中的小票字段,不合法则返回错误信息。

[0046] 基于上述实现用户登录的功能代码抽象出一个调用接口IPTLoginLib,该接口中拥有三个方法或函数,分别与上述(1)-(3)的功能对应,例如分别为:setUserInfo方法,handleVcodeImage方法和doPTLogin方法;

[0047] 该接口的调用顺序是:用户输入用户名密码,调用setUserInfo传入用户名和密码,然后调用handleVcodeImage,用传入的uin去服务器判断是否需要验证码,若需要验证码则调用Context中的显示验证码的回调函数传回验证码图片bitmap,若不需要验证码则直接从服务器返回获取vcode,获得vcode后,调用doPTLogin进行登录,并通过context把登录结果显示出来。然后程序中由此接口的实现类PTLoginLib实现IPTLoginLib中的三个方法,执行真正的登录逻辑。调用的地方都是使用IPTLoginLib接口调用,并不直接使用实现类PTLoginLib。

[0048] 当主程序已经发布,而需要对IPTLoginLib的实现类PTLoginLib进行升级时,需要定义新的用于完成更新的功能代码逻辑的实现类PTLoginLibV2,并将其置于更新库文件中,PTLoginLibV2依然继承主程序中的IPTLoginLib接口。在PTLoginLibV2中使用到的主程序中的代码都作为UserLib引用,在PTLoginLibV2编译的时候确保编译通过,但是编译完成的class文件中,不包含除PTLoginLibV2以外的任何代码。

[0049] 在之前的实现类PTLoginLib中,用到了接口定义IPTLoginLib、以及网络访问的方法,这些方法在PTLoginLibV2的实现中也需要用到,但是复制一份这些用到的类会导致把PTLoginLibV2加载到内存中时,与内存中原有的类发生重名的冲突。所以在编译PTLoginLibV2的时候,要把这些引用到的原来项目中的方法进行特殊处理,让PTLoginLibV2编译后的class文件,不包含重复的类。这个特殊处理过程,就是把原有的项目作为PTLoginLibV2的一个引用项目,但是在编译PTLoginLibV2的时候,在编译过程中配置选项过滤掉可能重复的类的数据,使得PTLoginLibV2.class文件中仅仅包含自己的逻辑代码,引用到的其他代码都是其引用项目中的。

[0050] 编译好PTLoginLibV2后,对已经发布的PTLoginLib进行升级,则需要配置更新配置服务器,让其向主程序返回新的版本控制信息,示例内容如下:

```
[0051]  {"v": "1",
[0052]   "s": "C19759F9268CD2383FA0F7469BC6188C",
[0053]   "url": "http://palm.qq.com/autoupdate/PTLoginLibV2.apk",
[0054]   "f": "PTLoginLib.apk",
[0055]   "c1": [{
[0056]     "p": "com.tencent.android.im.service.IPTLoginLib",
[0057]     "c": "com.tencent.android.im.service.PTLoginLibV2"
[0058]   }]
[0059] }
```

[0060] 版本控制信息以JSON形式下发,其字段含义意义如下:

[0061] v version当前最新配置信息版本

- [0062] s signature数据包签名
- [0063] url 最新更新库文件的网络地址
- [0064] f file更新库文件的名称
- [0065] cl class list,更新库文件中包含的实现类列表,其内容是JSONArray
- [0066] p 当前版本控制信息所指示的实现类所对应的调用接口名称
- [0067] c 当前版本控制信息所指示的实现类的名称
- [0068] 当主程序从网络获取上述版本控制信息后,与本地保存的版本控制信息进行比对,若不一致,即发现有更新时,则把本地版本控制信息更新到最新。然后根据url去网络下载最新的更新库文件,下载完成后计算更新库文件的签名,然后与版本控制信息中的数据包签名进行比对,比对签名符合,则把更新库文件加载到一个子classloader中同时把更新库文件写入本地,供以后调用。(若无更新,则直接读取本地版本控制信息,加载本地库文件中的实现类)。程序使用的时候,传入需要实例的调用接口名称(如com.tencent.android.im.service.IPTLoginLib),与版本控制信息中的p字段匹配,匹配成功以后,则用子classloader加载c字段所表示类实例,返回给调用的函数,供程序使用。
- [0069] 上述方法摆脱了android原生的appmarket所提供的方式的局限性。只更新所需要的代码即可,节约流量。
- [0070] 基于本发明提供的软件更新方法,本发明还提出一种软件更新装置,由于该装置是为了实现上述方法,所以该装置中的模块都是为实现本发明提出的软件更新方法中各步骤流程而设,为了节约篇幅,本发明装置实施例中省略了与前述方法实施例相同的部分。
- [0071] 图2为本发明软件升级装置的结构示意图,该装置200包括:
- [0072] 主程序201,用于通过调用接口加载实现类;
- [0073] 调用接口204,用于调用实现类;
- [0074] 实现类205,用于继承所述调用接口,实现具体的程序功能,与特定的功能代码对应;
- [0075] 当需要更新所述功能代码时,主程序获取版本的实现类,用新版本的实现类替换旧的实现类。
- [0076] 进一步地,所述主程序201进一步包含:
- [0077] 版本控制模块202,用于在主程序在启动时,从服务器获取版本控制信息,将其与本地的版本控制信息进行比对,判断所述功能代码是否需要更新,在需要更新时还负责下载包含新定义的实现类的更新库文件,用更新库文件替换本地保存的包含旧版本实现类的更新库文件并更新本地版本控制信息;
- [0078] 程序加载模块203,用于根据版本控制信息,通过调用接口204加载与功能代码对应的实现类。
- [0079] 优选地,版本控制模块202通过版本信息的比对发现所述功能代码有更新时,提示用户选择是否更新,在用户选择更新或执行静默更新的情况下,版本控制模块202才会执行后续操作;
- [0080] 所述程序加载模块203在加载更新库文件中的实现类失败时,加载主程序中设置默认的与所述功能代码对应的实现类。
- [0081] 优选地,本发明所述装置200应用于基于Android操作系统。

[0082] 以上所述,仅为本发明的较佳实施例而已,并非用于限定本发明保护范围。

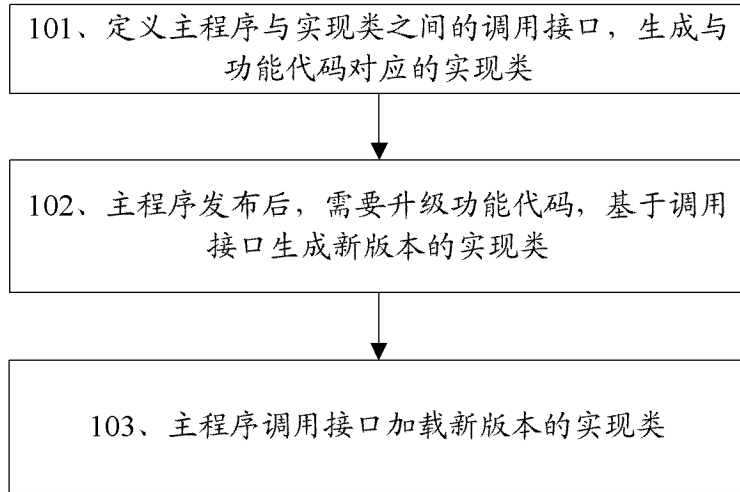


图1

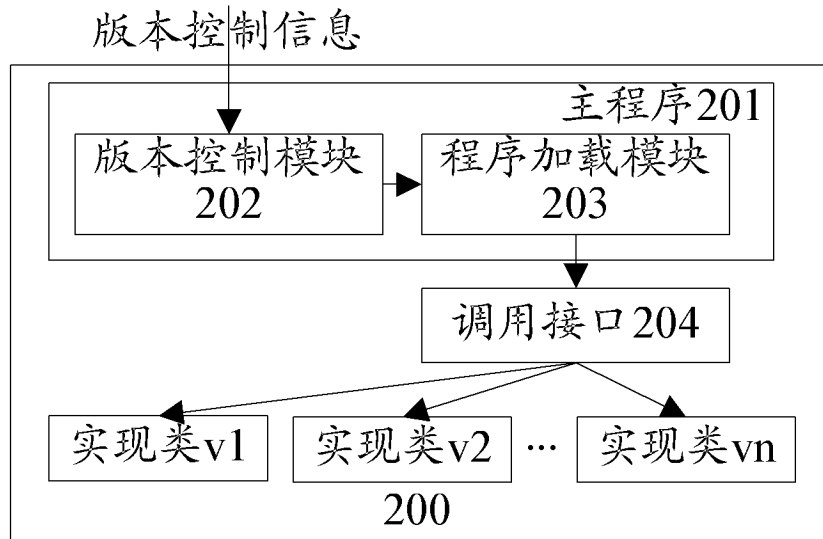


图2