(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 110209434 B
(45)授权公告日 2022.04.22

(21)申请号 201910330752.7

(22)申请日 2019.04.23

(65)同一申请的已公布的文献号
申请公布号 CN 110209434 A

(43)申请公布日 2019.09.06

(73)专利权人 努比亚技术有限公司
地址 518000 广东省深圳市南山区高新区北环大道9018号大族创新大厦A区10楼

(72)发明人 倪秉炬

(74)专利代理机构 深圳智汇远见知识产权代理有限公司 44481
代理人 左涛

(51)Int.Cl.
*G06F 9/445*(2018.01)
*G06F 8/65*(2018.01)

(56)对比文件
CN 103631730 A,2014.03.12
CN 105988874 A,2016.10.05
CN 103677923 A,2014.03.26
CN 108228496 A,2018.06.29
CN 104407852 A,2015.03.11
CN 103136003 A,2013.06.05
CN 108268328 A,2018.07.10
CN 105824700 A,2016.08.03
CN 101369232 A,2009.02.18
CN 101110044 A,2008.01.23
CN 104063239 A,2014.09.24
CN 1744046 A,2006.03.08
CN 101937398 A,2011.01.05
CN 102750157 A,2012.10.24
CN 103839000 A,2014.06.04
CN 102609305 A,2012.07.25
CN 104536764 A,2015.04.22
US 2011093701 A1,2011.04.21
CN 104375828 A,2015.02.25
US 2013137493 A1,2013.05.30
US 2015106147 A1,2015.04.16
CN 108595261 A,2018.09.28
CN 108255526 A,2018.07.06

审查员 李晓晖

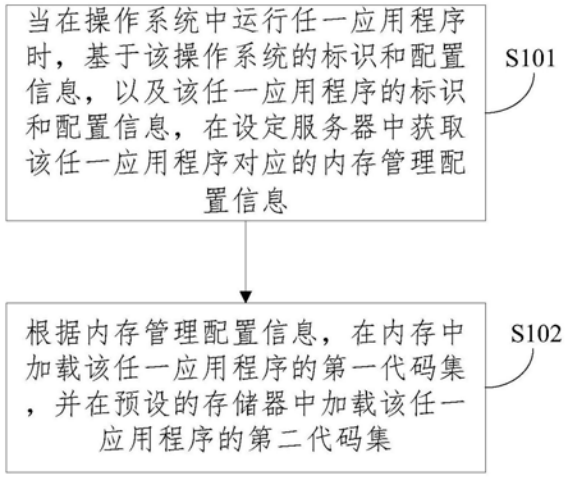
权利要求书2页 说明书14页 附图7页

(54)发明名称

一种内存管理方法、装置及计算机可读存储介质

(57)摘要

本申请涉及一种内存管理方法，包括：当在操作系统中运行任一应用程序时，基于所述操作系统的标识和配置信息，以及所述任一应用程序的标识和配置信息，在设定服务器中获取所述任一应用程序对应的内存管理配置信息；根据所述内存管理配置信息，在内存中加载所述任一应用程序的第一代码集，并在预设的存储器中加载所述任一应用程序的第二代码集。本发明还公开了一种内存管理装置及计算机可读存储介质，通过实施上述方案，实现了对内存进行精准管理，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。




CN 110209434 B

1.一种内存管理方法,其特征在于,包括:

当在操作系统中运行任一应用程序时,基于所述操作系统的标识和配置信息,以及所述任一应用程序的标识和配置信息,在设定服务器中获取所述任一应用程序对应的内存管理配置信息;

根据所述内存管理配置信息,在内存中加载所述任一应用程序的第一代码集,并在预设的存储器中加载所述任一应用程序的第二代码集;其中,所述内存管理配置信息包括:所述第一代码集的标识和所述第二代码集的标识;

当检测到所述操作系统的第一功能配置更改时,在所述第二代码集中确定所述第一功能配置对应的第三代码集;

将所述第三代码集由所述存储器中加载至所述内存中。

2.根据权利要求1所述的方法,其特征在于,当检测到所述操作系统的第一配置项更改时,所述方法还包括:

在所述第一代码集中确定所述第一功能配置对应的第四代码集;

将所述第四代码集由所述内存中加载至所述存储器中。

3.根据权利要求1所述的方法,其特征在于,在所述根据所述内存管理配置信息,在内存中加载所述任一应用程序的第一代码集,并在预设的存储器中加载所述任一应用程序的第二代码集之后,所述方法还包括:

当检测到所述任一应用程序的第二功能配置更改时,在所述第二代码集中确定所述第二功能配置对应的第五代码集;

将所述第五代码集由所述存储器中加载至所述内存中。

4.根据权利要求3所述的方法,其特征在于,当检测到所述任一应用程序的第二配置项更改时,所述方法还包括:

在所述第一代码集中确定所述第二功能配置对应的第六代码集;

将所述第六代码集由所述内存中加载至所述存储器中。

5.根据权利要求3所述的方法,其特征在于,在所述根据所述内存管理配置信息,在内存中加载所述任一应用程序的第一代码集,并在预设的存储器中加载所述任一应用程序的第二代码集之后,所述方法还包括:

当检测到所述任一应用程序版本更新时,基于所述操作系统的标识和配置信息,以及所述任一应用程序的当前标识和当前配置信息,在所述设定服务器中获取所述任一应用程序当前版本对应的内存管理配置信息;

根据所述任一应用程序当前版本对应的内存管理配置信息,在内存中加载所述任一应用程序的第七代码集,并在预设的存储器中加载所述任一应用程序的第八代码集。

6.一种内存管理方法,其特征在于,包括:

当在操作系统中运行任一应用程序时,在内存中加载所述任一应用程序的已运行功能组件对应的第一代码集,并在预设的存储器中加载所述任一应用程序的未运行功能组件对应的第二代码集,其中,所述第二代码集中包括与第一功能配置对应的第三代码集,所述第三代码集用于当检测到所述操作系统的第一功能配置更改时由所述存储器中加载至所述内存中;

将内存管理配置信息上传至设定服务器;其中,所述内存管理配置信息包括:所述操作

系统的标识和配置信息，所述任一应用程序的标识和配置信息，以及所述第一代码集的标识和所述第二代码集的标识。

7.根据权利要求6所述的方法，其特征在于，在监测到所述任一应用程序的版本已更新的情况下，所述方法还包括：

当在所述操作系统中运行版本已更新的所述任一应用程序时，在内存中加载版本已更新的所述任一应用程序已运行功能组件对应的第九代码集，并在预设的存储器中加载版本已更新的所述任一应用程序未运行功能组件对应的第十代码集；

将当前内存管理配置信息上传至设定服务器；其中，所述当前内存管理配置信息包括：所述操作系统的标识和配置信息，所述任一应用程序的当前标识和当前配置信息，以及所述第九代码集的标识和所述第十代码集的标识。

8.一种内存管理装置，其特征在于，所述内存管理装置包括：

存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序；

所述计算机程序被所述处理器执行时实现如权利要求1至5中任一项所述的内存管理方法的步骤，和/或如权利要求6至7中任一项所述的内存管理方法的步骤。

9.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质上存储有内存管理程序，所述内存管理程序被处理器执行时实现如权利要求1至5中任一项所述的内存管理方法的步骤，和/或如权利要求6至7中任一项所述的内存管理方法的步骤。

# 一种内存管理方法、装置及计算机可读存储介质

## 技术领域

[0001] 本申请涉及可移动终端领域，尤其涉及一种内存管理方法、装置及计算机可读存储介质。

## 背景技术

[0002] 随着计算机设备(例如，移动终端)的快速发展，计算机设备中安装的应用程序(App，Application)也得到了快速普及。随着应用程序的功能丰富，应用程序的容量也越来越大。应用程序的开发人员在开发应用程序时，通常不会考虑在计算机设备中使用应用程序时，应用程序加载到计算机设备内存的优化，导致在内存中加载应用程序时，会在内存中加载大量的应用程序代码分支，占用大量的计算机设备内存，增加计算机设备的电量消耗的同时，也降低了计算机设备的运行速度。

## 发明内容

[0003] 为了解决上述技术问题或者至少部分地解决上述技术问题，本申请提供了一种内存管理方法、装置及计算机可读存储介质。

[0004] 第一方面，本申请提供了一种内存管理方法，包括：

[0005] 当在操作系统中运行任一应用程序时，基于所述操作系统的标识和配置信息，以及所述任一应用程序的标识和配置信息，在设定服务器中获取所述任一应用程序对应的内存管理配置信息；

[0006] 根据所述内存管理配置信息，在内存中加载所述任一应用程序的第一代码集，并在预设的存储器中加载所述任一应用程序的第二代码集；其中，所述内存管理配置信息包括：所述第一代码集的标识和所述第二代码集的标识。

[0007] 可选地，在所述根据所述内存管理配置信息，在内存中加载所述任一应用程序的第一代码集，并在预设的存储器中加载所述任一应用程序的第二代码集之后，所述方法还包括：

[0008] 当检测到所述操作系统的第一功能配置更改时，在所述第二代码集中确定所述第一功能配置对应的第三代码集；

[0009] 将所述第三代码集由所述存储器中加载至所述内存中。

[0010] 可选地，当检测到所述操作系统的第一配置项更改时，所述方法还包括：

[0011] 在所述第一代码集中确定所述第一功能配置对应的第四代码集；

[0012] 将所述第四代码集由所述内存中加载至所述存储器中。

[0013] 可选地，在所述根据所述内存管理配置信息，在内存中加载所述任一应用程序的第一代码集，并在预设的存储器中加载所述任一应用程序的第二代码集之后，所述方法还包括：

[0014] 当检测到所述任一应用程序的第二功能配置更改时，在所述第二代码集中确定所述第二功能配置对应的第五代码集；

[0015] 将所述第五代码集由所述存储器中加载至所述内存中。

[0016] 可选地，当检测到所述任一应用程序的第二配置项更改时，所述方法还包括：

[0017] 在所述第一代码集中确定所述第二功能配置对应的第六代码集；

[0018] 将所述第六代码集由所述内存中加载至所述存储器中。

[0019] 可选地，在所述根据所述内存管理配置信息，在内存中加载所述任一应用程序的第一代码集，并在预设的存储器中加载所述任一应用程序的第二代码集之后，所述方法还包括：

[0020] 当检测到所述任一应用程序版本更新时，基于所述操作系统的标识和配置信息，以及所述任一应用程序的当前标识和当前配置信息，在所述设定服务器中获取所述任一应用程序当前版本对应的内存管理配置信息；

[0021] 根据所述任一应用程序当前版本对应的内存管理配置信息，在内存中加载所述任一应用程序的第七代码集，并在预设的存储器中加载所述任一应用程序的第八代码集。

[0022] 第二方面，本申请提供了一种内存管理方法，包括：

[0023] 当在操作系统中运行任一应用程序时，在内存中加载所述任一应用程序的已运行功能组件对应的第一代码集，并在预设的存储器中加载所述任一应用程序的未运行功能组件对应的第二代码集；

[0024] 将内存管理配置信息上传至设定服务器；其中，所述内存管理配置信息包括：所述操作系统的标识和配置信息，所述任一应用程序的标识和配置信息，以及所述第一代码集的标识和所述第二代码集的标识。

[0025] 可选地，在监测到所述任一应用程序的版本已更新的情况下，所述方法还包括：

[0026] 当在所述操作系统中运行版本已更新的所述任一应用程序时，在内存中加载版本已更新的所述任一应用程序已运行功能组件对应的第九代码集，并在预设的存储器中加载版本已更新的所述任一应用程序未运行功能组件对应的第十代码集；

[0027] 将当前内存管理配置信息上传至设定服务器；其中，所述当前内存管理配置信息包括：所述操作系统的标识和配置信息，所述任一应用程序的当前标识和当前配置信息，以及所述第九代码集的标识和所述第十代码集的标识。

[0028] 第三方面，本申请提供了一种内存管理装置，所述内存管理装置包括：

[0029] 存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序；

[0030] 所述计算机程序被所述处理器执行时实现上述的内存管理方法的步骤。

[0031] 第四方面，本申请提供了一种计算机可读存储介质，所述计算机可读存储介质上存储有内存管理程序，所述内存管理程序被处理器执行时实现上述的内存管理方法的步骤。

[0032] 本申请实施例提供的上述技术方案与现有技术相比具有如下优点：

[0033] 本申请实施例提供的该方法，通过对计算机设备中应用程序在运行时的内存加载代码集的监控，并将监控得到的内存管理配置信息上传至服务器，以供其他计算机设备根据在服务器中获取的内存管理配置信息，对内存进行精准管理，避免了在内存中加载过多的应用程序代码集，导致内存占用过多，导致计算机终端运行缓慢，以及耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

## 附图说明

[0034] 此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本发明的实施例，并与说明书一起用于解释本发明的原理。

[0035] 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地介绍，显而易见地，对于本领域普通技术人员而言，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。

[0036] 图1为本申请各个实施例提供的移动终端的一种示意图；

[0037] 图2为本发明实施例提供的一种通信网络系统架构图；

[0038] 图3为本发明第一实施例所述的在操作系统中运行任一应用程序的示意图；

[0039] 图4为本发明第一实施例所述的内存管理方法流程图；

[0040] 图5为本发明第二实施例所述的内存管理方法流程图；

[0041] 图6为本发明第三实施例所述的内存管理方法流程图；

[0042] 图7为本发明第四实施例所述的内存管理方法流程图；

[0043] 图8为本发明第五实施例所述的内存管理方法流程图；

[0044] 图9为本发明第六实施例所述的内存管理装置组成结构示意图。

## 具体实施方式

[0045] 应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0046] 在后续的描述中，使用用于表示元件的诸如“模块”、“部件”或“单元”的后缀仅为了有利于本发明的说明，其本身没有特定的意义。因此，“模块”、“部件”或“单元”可以混合地使用。

[0047] 终端可以以各种形式来实施。例如，本发明中描述的终端可以包括诸如手机、平板电脑、笔记本电脑、掌上电脑、个人数字助理(Personal Digital Assistant，PDA)、便捷式媒体播放器(Portable Media Player，PMP)、导航装置、可穿戴设备、智能手环、计步器等移动终端，以及诸如数字TV、台式计算机等固定终端。

[0048] 后续描述中将以移动终端为例进行说明，本领域技术人员将理解的是，除了特别用于移动目的的元件之外，根据本发明的实施方式的构造也能够应用于固定类型的终端。

[0049] 请参阅图1，其为实现本发明各个实施例的一种移动终端的硬件结构示意图，该移动终端100可以包括：RF(Radio Frequency，射频)单元101、WiFi模块102、音频输出单元103、A/V(音频/视频)输入单元104、传感器105、显示单元106、用户输入单元107、接口单元108、存储器109、处理器110、以及电源111等部件。本领域技术人员可以理解，图1中示出的移动终端结构并不构成对移动终端的限定，移动终端可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件布置。

[0050] 下面结合图1对移动终端的各个部件进行具体的介绍：

[0051] 射频单元101可用于收发信息或通话过程中，信号的接收和发送，具体的，将基站的下行信息接收后，给处理器110处理；另外，将上行的数据发送给基站。通常，射频单元101包括但不限于天线、至少一个放大器、收发信机、耦合器、低噪声放大器、双工器等。此外，射频单元101还可以通过无线通信与网络和其他设备通信。上述无线通信可以使用任一通信标准或协议，包括但不限于GSM(Global System of Mobile communication，全球移动通讯

系统)、GPRS(General Packet Radio Service,通用分组无线服务)、CDMA2000(Code Division Multiple Access 2000,码分多址2000)、WCDMA(Wideband Code Division Multiple Access,宽带码分多址)、TD-SCDMA(Time Division-Synchronous Code Division Multiple Access,时分同步码分多址)、FDD-LTE(Frequency Division Duplexing-Long Term Evolution,频分双工长期演进)和TDD-LTE(Time Division Duplexing-Long Term Evolution,分时双工长期演进)等。

[0052] WiFi属于短距离无线传输技术,移动终端通过WiFi模块102可以帮助用户收发电子邮件、浏览网页和访问流式媒体等,它为用户提供了无线的宽带互联网访问。虽然图1示出了WiFi模块102,但是可以理解的是,其并不属于移动终端的必须构成,完全可以根据需要在不改变发明的本质的范围内而省略。

[0053] 音频输出单元103可以在移动终端100处于呼叫信号接收模式、通话模式、记录模式、语音识别模式、广播接收模式等等模式下时,将射频单元101或WiFi模块102接收的或者在存储器109中存储的音频数据转换成音频信号并且输出为声音。而且,音频输出单元103还可以提供与移动终端100执行的特定功能相关的音频输出(例如,呼叫信号接收声音、消息接收声音等等)。音频输出单元103可以包括扬声器、蜂鸣器等等。

[0054] A/V输入单元104用于接收音频或视频信号。A/V输入单元104可以包括图形处理器(Graphics Processing Unit,GPU)1041和麦克风1042,图形处理器1041对在视频捕获模式或图像捕获模式中由图像捕获装置(如摄像头)获得的静态图片或视频的图像数据进行处理。处理后的图像帧可以显示在显示单元106上。经图形处理器1041处理后的图像帧可以存储在存储器109(或其它存储介质)中或者经由射频单元101或WiFi模块102进行发送。麦克风1042可以在电话通话模式、记录模式、语音识别模式等等运行模式中经由麦克风1042接收声音(音频数据),并且能够将这样的声音处理为音频数据。处理后的音频(语音)数据可以在电话通话模式的情况下转换为可经由射频单元101发送到移动通信基站的格式输出。麦克风1042可以实施各种类型的噪声消除(或抑制)算法以消除(或抑制)在接收和发送音频信号的过程中产生的噪声或者干扰。

[0055] 移动终端100还包括至少一种传感器105,比如光传感器、运动传感器以及其他传感器。具体地,光传感器包括环境光传感器及接近传感器,其中,环境光传感器可根据环境光线的明暗来调节显示面板1061的亮度,接近传感器可在移动终端100移动到耳边时,关闭显示面板1061和/或背光。作为运动传感器的一种,加速计传感器可检测各个方向上(一般为三轴)加速度的大小,静止时可检测出重力的大小及方向,可用于识别手机姿态的应用(比如横竖屏切换、相关游戏、磁力计姿态校准)、振动识别相关功能(比如计步器、敲击)等;至于手机还可配置的指纹传感器、压力传感器、虹膜传感器、分子传感器、陀螺仪、气压计、湿度计、温度计、红外线传感器等其他传感器,在此不再赘述。

[0056] 显示单元106用于显示由用户输入的信息或提供给用户的信息。显示单元106可包括显示面板1061,可以采用液晶显示器(Liquid Crystal Display,LCD)、有机发光二极管(Organic Light-Emitting Diode,OLED)等形式来配置显示面板1061。

[0057] 用户输入单元107可用于接收输入的数字或字符信息,以及产生与移动终端的用户设置以及功能控制有关的键信号输入。具体地,用户输入单元107可包括触控面板1071以及其他输入设备1072。触控面板1071,也称为触摸屏,可收集用户在其上或附近的触摸操作

(比如用户使用手指、触笔等任何适合的物体或附件在触控面板1071上或在触控面板1071附近的操作),并根据预先设定的程式驱动相应的连接装置。触控面板1071可包括触摸检测装置和触摸控制器两个部分。其中,触摸检测装置检测用户的触摸方位,并检测触摸操作带来的信号,将信号传送给触摸控制器;触摸控制器从触摸检测装置上接收触摸信息,并将它转换成触点坐标,再送给处理器110,并能接收处理器110发来的命令并加以执行。此外,可以采用电阻式、电容式、红外线以及表面声波等多种类型实现触控面板1071。除了触控面板1071,用户输入单元107还可以包括其他输入设备1072。具体地,其他输入设备1072可以包括但不限于物理键盘、功能键(比如音量控制按键、开关按键等)、轨迹球、鼠标、操作杆等中的一种或多种,具体此处不做限定。

[0058] 进一步的,触控面板1071可覆盖显示面板1061,当触控面板1071检测到在其上或附近的触摸操作后,传送给处理器110以确定触摸事件的类型,随后处理器110根据触摸事件的类型在显示面板1061上提供相应的视觉输出。虽然在图1中,触控面板1071与显示面板1061是作为两个独立的部件来实现移动终端的输入和输出功能,但是在某些实施例中,可以将触控面板1071与显示面板1061集成而实现移动终端的输入和输出功能,具体此处不做限定。

[0059] 接口单元108用作至少一个外部装置与移动终端100连接可以通过的接口。例如,外部装置可以包括有线或无线头戴式耳机端口、外部电源(或电池充电器)端口、有线或无线数据端口、存储卡端口、用于连接具有识别模块的装置的端口、音频输入/输出(I/O)端口、视频I/O端口、耳机端口等等。接口单元108可以用于接收来自外部装置的输入(例如,数据信息、电力等等)并且将接收到的输入传输到移动终端100内的一个或多个元件或者可以用于在移动终端100和外部装置之间传输数据。

[0060] 存储器109可用于存储软件程序以及各种数据。存储器109可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序(比如声音播放功能、图像播放功能等)等;存储数据区可存储根据手机的使用所创建的数据(比如音频数据、电话本等)等。此外,存储器109可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。

[0061] 处理器110是移动终端的控制中心,利用各种接口和线路连接整个移动终端的各个部分,通过运行或执行存储在存储器109内的软件程序和/或模块,以及调用存储在存储器109内的数据,执行移动终端的各种功能和处理数据,从而对移动终端进行整体监控。处理器110可包括一个或多个处理单元;优选的,处理器110可集成应用处理器和调制解调处理器,其中,应用处理器主要处理操作系统、用户界面和应用程序等,调制解调处理器主要处理无线通信。可以理解的是,上述调制解调处理器也可以不集成到处理器110中。

[0062] 移动终端100还可以包括给各个部件供电的电源111(比如电池),优选的,电源111可以通过电源管理系统与处理器110逻辑相连,从而通过电源管理系统实现管理充电、放电、以及功耗管理等功能。

[0063] 尽管图1未示出,移动终端100还可以包括蓝牙模块等,在此不再赘述。

[0064] 为了便于理解本发明实施例,下面对本发明的移动终端所基于的通信网络系统进行描述。

[0065] 请参阅图2,图2为本发明实施例提供的一种通信网络系统架构图,该通信网络系

统为通用移动通信技术的LTE系统，该LTE系统包括依次通讯连接的UE(User Equipment，用户设备)201，E-UTRAN(Evolved UMTS Terrestrial Radio Access Network，演进式UMTS陆地无线接入网)202，EPC(Evolved Packet Core，演进式分组核心网)203和运营商的IP业务204。

[0066] 具体地，UE201可以是上述终端100，此处不再赘述。

[0067] E-UTRAN202包括eNodeB2021和其它eNodeB2022等。其中，eNodeB2021可以通过回程(backhaul)(例如X2接口)与其它eNodeB2022连接，eNodeB2021连接到EPC203，eNodeB2021可以提供UE201到EPC203的接入。

[0068] EPC203可以包括MME(Mobility Management Entity，移动性管理实体)2031，HSS(Home Subscriber Server，归属用户服务器)2032，其它MME2033，SGW(Serving Gate Way，服务网关)2034，PGW(PDN Gate Way，分组数据网络网关)2035和PCRF(Policy and Charging Rules Function，政策和资费功能实体)2036等。其中，MME2031是处理UE201和EPC203之间信令的控制节点，提供承载和连接管理。HSS2032用于提供一些寄存器来管理诸如归属位置寄存器(图中未示)之类的功能，并且保存有一些有关服务特征、数据速率等用户专用的信息。所有用户数据都可以通过SGW2034进行发送，PGW2035可以提供UE 201的IP地址分配以及其它功能，PCRF2036是业务数据流和IP承载资源的策略与计费控制策略决策点，它为策略与计费执行功能单元(图中未示)选择及提供可用的策略和计费控制决策。

[0069] IP业务204可以包括因特网、内联网、IMS(IP Multimedia Subsystem，IP多媒体子系统)或其它IP业务等。

[0070] 虽然上述以LTE系统为例进行了介绍，但本领域技术人员应当知晓，本发明不仅仅适用于LTE系统，也可以适用于其他无线通信系统，例如GSM、CDMA2000、WCDMA、TD-SCDMA以及未来新的网络系统等，此处不做限定。

[0071] 基于上述移动终端硬件结构以及通信网络系统，提出本发明方法各个实施例。

[0072] 本发明第一实施例，一种内存管理方法，如图3～图4所示，包括以下具体步骤：

[0073] 步骤S101，当在操作系统中运行任一应用程序时，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中获取该任一应用程序对应的内存管理配置信息。

[0074] 在本实施例中，对操作系统(OS，Operating System)不做具体限定，可以是微软(Microsoft)公司的视窗(Windows)操作系统，也可以是苹果(Apple)公司的IOS操作系统或者MAC OS操作系统，也可以是谷歌(google)公司的安卓(Android)操作系统，也可以是Unix操作系统，也可以是Linux操作系统，也可以是基于上述系统的再开发的改进操作系统；同时，在本实施例中，对操作系统的版本也不做具体限定。

[0075] 在本实施例中，操作系统的标识包括以下标识中的一种或多种的组合：操作系统的类型标识，操作系统的名称标识及操作系统的版本标识等；操作系统的配置信息包括以下配置信息中的一种或多种的组合：操作系统中与该任一应用程序中功能组件相关的功能组件配置信息，以及操作系统中与该任一应用程序中功能组件相关的硬件模块配置信息。

[0076] 在本实施例中，该任一应用程序的标识包括以下标识中的一种或多种的组合：应用程序的类型标识，应用程序的版本标识，及应用程序的包标识等；应用程序的配置信息包括：应用程序的功能组件配置信息等。

[0077] 在本实施例中，设定服务器中获取该任一应用程序对应的内存管理配置信息的方式包括但不限于以下方式之一：

[0078] 方式一，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，向设定服务器发送内存管理配置信息请求，以供设定服务器查询与内存管理配置信息请求匹配的该任一应用程序对应的内存管理配置信息；接收设定服务器反馈的该任一应用程序对应的内存管理配置信息；其中，内存管理配置信息请求，包括：该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息；

[0079] 方式二，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中查询并读取已查询到的该任一应用程序对应的内存管理配置信息；其中，该任一应用程序对应的内存管理配置信息与该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息匹配。

[0080] 步骤S102，根据内存管理配置信息，在内存中加载该任一应用程序的第一代码集，并在预设的存储器中加载该任一应用程序的第二代码集。

[0081] 其中，内存管理配置信息包括：第一代码集的标识和第二代码集的标识。

[0082] 在本实施例中，对预设的存储器不做具体限定，可以是计算机设备(例如，移动终端)的存储器中的设定托管空间。

[0083] 在本实施例中，第一代码集的标识包括：应用程序的当前调用的所有第一功能组件对应的所有分支代码；第二代码集的标识包括：应用程序当前并未调用的所有第二功能组件对应的所有分支代码。

[0084] 本发明第一实施例所述的内存管理方法，实现了对内存的精准管理，避免了在内存中加载过多的应用程序代码集，导致内存占用过多，导致计算机终端运行缓慢，以及耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0085] 本发明第二实施例，一种内存管理方法，如图5所示，包括以下具体步骤：

[0086] 步骤S201，当在操作系统中运行任一应用程序时，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中获取该任一应用程序对应的内存管理配置信息。

[0087] 在本实施例中，对操作系统(OS，Operating System)不做具体限定，可以是微软(Microsoft)公司的视窗(Windows)操作系统，也可以是苹果(Apple)公司的IOS操作系统或者MAC OS操作系统，也可以是谷歌(google)公司的安卓(Android)操作系统，也可以是Unix操作系统，也可以是Linux操作系统，也可以是基于上述系统的再开发的改进操作系统；同时，在本实施例中，对操作系统的版本也不做具体限定。

[0088] 在本实施例中，操作系统的标识包括以下标识中的一种或多种的组合：操作系统的类型标识，操作系统的名称标识及操作系统的版本标识等；操作系统的配置信息包括以下配置信息中的一种或多种的组合：操作系统中与该任一应用程序中功能组件相关的功能组件配置信息，以及操作系统中与该任一应用程序中功能组件相关的硬件模块配置信息。

[0089] 在本实施例中，该任一应用程序的标识包括以下标识中的一种或多种的组合：应用程序的类型标识，应用程序的版本标识，及应用程序的包标识等；应用程序的配置信息包括：应用程序的功能组件配置信息等。

[0090] 在本实施例中，设定服务器中获取该任一应用程序对应的内存管理配置信息的方

式包括但不限于以下方式之一：

[0091] 方式一，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，向设定服务器发送内存管理配置信息请求，以供设定服务器查询与内存管理配置信息请求匹配的该任一应用程序对应的内存管理配置信息；接收设定服务器反馈的该任一应用程序对应的内存管理配置信息；其中，内存管理配置信息请求，包括：该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息；

[0092] 方式二，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中查询并读取已查询到的该任一应用程序对应的内存管理配置信息；其中，该任一应用程序对应的内存管理配置信息与该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息匹配。

[0093] 步骤S202，根据内存管理配置信息，在内存中加载该任一应用程序的第一代码集，并在预设的存储器中加载该任一应用程序的第二代码集。

[0094] 其中，内存管理配置信息包括：第一代码集的标识和第二代码集的标识。

[0095] 在本实施例中，对预设的存储器不做具体限定，可以是计算机设备(例如，移动终端)的存储器中的设定托管空间。

[0096] 在本实施例中，第一代码集的标识包括：应用程序的当前调用的所有第一功能组件对应的所有分支代码；第二代码集的标识包括：应用程序当前并未调用的所有第二功能组件对应的所有分支代码。

[0097] 步骤S203，当检测到操作系统的第一功能配置更改时，在第二代码集中确定第一功能配置对应的第三代码集；将第三代码集由存储器中加载至内存中。

[0098] 在本实施例中，第三代码集的标识包括：第二代码集中与操作系统的第一功能配置相关的所有分支代码；

[0099] 当检测到操作系统的第一功能配置更改时，在第二代码集中确定第一功能配置对应的第三代码集；将第三代码集由存储器中实时加载至内存中，保证了应用程序的正常运行，避免了内存中未加载第三代码集导致应用程序出错，甚至导致应用程序卡死的弊端。

[0100] 可选地，该内存管理方法还可以包括步骤S204，步骤S204具体包括：当检测到操作系统的第一功能配置更改时，在第一代码集中确定第一功能配置对应的第四代码集；将第四代码集由内存中加载至存储器中。

[0101] 其中，对步骤S204的执行顺序不做具体限定，可以在步骤S202之后执行，也可以在步骤S203之后执行，也可以与步骤S203同时执行。

[0102] 在本实施例中，第四代码集的标识包括：第二代码集中与操作系统的第一功能配置相关的所有分支代码；

[0103] 当检测到操作系统的第一功能配置更改时，在第一代码集中确定第一功能配置对应的第四代码集；将第四代码集由内存中加载至存储器中，有效实现了在操作系统的功能配置更改时，将与操作系统的功能配置无关的功能配置转移至存储器中，避免了在内存中加载与应用程序当前无需使用的功能组件对应的代码集，导致内存加载过多代码，进而导致计算机设备运行缓慢，耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0104] 本发明第二实施例所述的内存管理方法，实现了对内存的精准管理，避免了在内

存中加载过多的应用程序代码集，导致内存占用过多，导致计算机终端运行缓慢，以及耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0105] 本发明第三实施例，一种内存管理方法，如图6所示，包括以下具体步骤：

[0106] 步骤S301，当在操作系统中运行任一应用程序时，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中获取该任一应用程序对应的内存管理配置信息。

[0107] 在本实施例中，对操作系统(OS，Operating System)不做具体限定，可以是微软(Microsoft)公司的视窗(Windows)操作系统，也可以是苹果(Apple)公司的IOS操作系统或者MAC OS操作系统，也可以是谷歌(google)公司的安卓(Android)操作系统，也可以是Unix操作系统，也可以是Linux操作系统，也可以是基于上述系统的再开发的改进操作系统；同时，在本实施例中，对操作系统的版本也不做具体限定。

[0108] 在本实施例中，操作系统的标识包括以下标识中的一种或多种的组合：操作系统的类型标识，操作系统的名称标识及操作系统的版本标识等；操作系统的配置信息包括以下配置信息中的一种或多种的组合：操作系统中与该任一应用程序中功能组件相关的功能组件配置信息，以及操作系统中与该任一应用程序中功能组件相关的硬件模块配置信息。

[0109] 在本实施例中，该任一应用程序的标识包括以下标识中的一种或多种的组合：应用程序的类型标识，应用程序的版本标识，及应用程序的包标识等；应用程序的配置信息包括：应用程序的功能组件配置信息等。

[0110] 在本实施例中，设定服务器中获取该任一应用程序对应的内存管理配置信息的方式包括但不限于以下方式之一：

[0111] 方式一，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，向设定服务器发送内存管理配置信息请求，以供设定服务器查询与内存管理配置信息请求匹配的该任一应用程序对应的内存管理配置信息；接收设定服务器反馈的该任一应用程序对应的内存管理配置信息；其中，内存管理配置信息请求，包括：该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息；

[0112] 方式二，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中查询并读取已查询到的该任一应用程序对应的内存管理配置信息；其中，该任一应用程序对应的内存管理配置信息与该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息匹配。

[0113] 步骤S302，根据内存管理配置信息，在内存中加载该任一应用程序的第一代码集，并在预设的存储器中加载该任一应用程序的第二代码集。

[0114] 其中，内存管理配置信息包括：第一代码集的标识和第二代码集的标识。

[0115] 在本实施例中，对预设的存储器不做具体限定，可以是计算机设备(例如，移动终端)的存储器中的设定托管空间。

[0116] 在本实施例中，第一代码集的标识包括：应用程序的当前调用的所有第一功能组件对应的所有分支代码；第二代码集的标识包括：应用程序当前并未调用的所有第二功能组件对应的所有分支代码。

[0117] 步骤S303，当检测到该任一应用程序的第二功能配置更改时，在第二代码集中确定第二功能配置对应的第五代码集；将第五代码集由存储器中加载至内存中。

[0118] 在本实施例中，第五代码集的标识包括：第一代码集中与该任一应用程序的第二功能配置相关的所有分支代码。

[0119] 当检测到该任一应用程序的第二功能配置更改时，在第二代码集中确定第二功能配置对应的第五代码集；将第五代码集由存储器中实时加载至内存中，保证了应用程序在调用功能组件的正常运行，避免了内存中未加载第五代码集导致应用程序的调用功能组件出错，甚至导致应用程序卡死的弊端。

[0120] 可选地，该内存管理方法还可以包括步骤S304；步骤S304具体包括：当检测到该任一应用程序的第二配置项更改时，在第一代码集中确定第二功能配置对应的第六代码集；将第六代码集由内存中加载至存储器中。

[0121] 在本实施例中，第六代码集的标识包括：第一代码集中与该任一应用程序的第二功能配置相关的所有分支代码。

[0122] 其中，对步骤S304的执行顺序不做具体限定，可以在步骤S302之后执行，也可以在步骤S303之后执行，也可以与步骤S303同时执行。

[0123] 当检测到该任一应用程序的第二配置项更改时，在在第一代码集中确定第二功能配置对应的第六代码集；将第六代码集由内存中加载至存储器中，有效实现了在应用程序的功能配置更改时，将与操作系统的功能配置无关的功能配置转移至存储器中，避免了在内存中加载与应用程序当前无需使用的功能组件对应的代码集，导致内存加载过多代码，进而导致计算机设备运行缓慢，耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0124] 可选地，该内存管理方法还可以包括步骤S305；步骤S305具体包括：当检测到该任一应用程序版本更新时，基于操作系统的标识和配置信息，以及该任一应用程序的当前标识和当前配置信息，在设定服务器中获取该任一应用程序当前版本对应的内存管理配置信息；根据该任一应用程序当前版本对应的内存管理配置信息，在内存中加载该任一应用程序的第七代码集，并在预设的存储器中加载该任一应用程序的第八代码集。

[0125] 在本实施例中，第七代码集的标识包括：当前版本应用程序的当前调用的所有第三功能组件对应的所有分支代码；第八代码集的标识包括：当前版本应用程序当前并未调用的所有第四功能组件对应的所有分支代码。

[0126] 其中，对步骤S305的执行顺序不做具体限定，可以在步骤S302之后执行，也可以在步骤S303之后执行，也可以在步骤S304之后执行。

[0127] 当检测到该任一应用程序版本更新时，基于操作系统的标识和配置信息，以及该任一应用程序的当前标识和当前配置信息，在设定服务器中获取该任一应用程序当前版本对应的内存管理配置信息；根据该任一应用程序当前版本对应的内存管理配置信息，在内存中加载该任一应用程序的第七代码集，并在预设的存储器中加载该任一应用程序的第八代码集，实现了版本已更新的应用程序的内存管理配置信息的实时更新，有效避免了应用程序版本更新导致功能组件增加，功能组件减少，和/或功能组件对应的代码集更改，导致内存无法加载正确的代码集，导致应用程序无法正常运行，甚至应用程序崩溃的弊端。

[0128] 本发明第三实施例所述的内存管理方法，实现了对内存的精准管理，避免了在内存中加载过多的应用程序代码集，导致内存占用过多，导致计算机终端运行缓慢，以及耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0129] 本发明第四实施例，一种内存管理方法，如图7所示，包括以下具体步骤：

[0130] 步骤S401，当在操作系统中运行任一应用程序时，在内存中加载该任一应用程序的已运行功能组件对应的第一代码集，并在预设的存储器中加载该任一应用程序的未运行功能组件对应的第二代码集。

[0131] 在本实施例中，对预设的存储器不做具体限定，可以是计算机设备(例如，移动终端)的存储器中的设定托管空间。

[0132] 通过在内存中加载该任一应用程序的已运行功能组件对应的第一代码集，并在预设的存储器中加载该任一应用程序的未运行功能组件对应的第二代码集，有效避免了在内存中加载该任一应用程序中的所有功能组件中对应的代码集，导致在内存中加载过多的代码，导致计算机设备运行缓慢，耗电量过大的弊端，实现了对内存的精准管理。

[0133] 步骤S402，将内存管理配置信息上传至设定服务器。

[0134] 其中，内存管理配置信息包括：操作系统的标识和配置信息，任一应用程序的标识和配置信息，以及第一代码集的标识和第二代码集的标识。

[0135] 在本实施例中，对操作系统(OS，Operating System)不做具体限定，可以是微软(Microsoft)公司的视窗(Windows)操作系统，也可以是苹果(Apple)公司的IOS操作系统或者MAC OS操作系统，也可以是谷歌(google)公司的安卓(Android)操作系统，也可以是Unix操作系统，也可以是Linux操作系统，也可以是基于上述系统的再开发的改进操作系统；同时，在本实施例中，对操作系统的版本也不做具体限定。

[0136] 在本实施例中，操作系统的标识包括以下标识中的一种或多种的组合：操作系统的类型标识，操作系统的名称标识及操作系统的版本标识等；操作系统的配置信息包括以下配置信息中的一种或多种的组合：操作系统中与该任一应用程序中功能组件相关的功能组件配置信息，以及操作系统中与该任一应用程序中功能组件相关的硬件模块配置信息。

[0137] 在本实施例中，该任一应用程序的标识包括以下标识中的一种或多种的组合：应用程序的类型标识，应用程序的版本标识，及应用程序的包标识等；应用程序的配置信息包括：应用程序的功能组件配置信息等。

[0138] 通过对本计算机设备中应用程序中代码集在内存中的加载状态实时监控，并实时将包含应用程序中代码集在内存中的加载状态实时监控信息的内存管理配置信息上传至服务器，以供服务器将内存管理配置信息共享至其他计算机设备，有效提高了其他计算机设备的内存管理精准度，有效保证了其他计算机设备的运行速度，并有效降低了其他计算机设备的运行电量。

[0139] 本发明第四实施例所述的内存管理方法，实现了对内存的精准管理，避免了在内存中加载过多的应用程序代码集，导致内存占用过多，导致计算机终端运行缓慢，以及耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0140] 本发明第五实施例，一种内存管理方法，如图8所示，包括以下具体步骤：

[0141] 步骤S501，当在操作系统中运行任一应用程序时，在内存中加载该任一应用程序的已运行功能组件对应的第一代码集，并在预设的存储器中加载该任一应用程序的未运行功能组件对应的第二代码集。

[0142] 在本实施例中，对预设的存储器不做具体限定，可以是计算机设备(例如，移动终端)的存储器中的设定托管空间。

[0143] 通过在内存中加载该任一应用程序的已运行功能组件对应的第一代码集，并在预设的存储器中加载该任一应用程序的未运行功能组件对应的第二代码集，有效避免了在内存中加载该任一应用程序中的所有功能组件中对应的代码集，导致在内存中加载过多的代码，导致计算机设备运行缓慢，耗电量过大的弊端，实现了对内存的精准管理。

[0144] 步骤S502，将内存管理配置信息上传至设定服务器。

[0145] 其中，内存管理配置信息包括：操作系统的标识和配置信息，任一应用程序的标识和配置信息，以及第一代码集的标识和第二代码集的标识。

[0146] 在本实施例中，对操作系统(OS，Operating System)不做具体限定，可以是微软(Microsoft)公司的视窗(Windows)操作系统，也可以是苹果(Apple)公司的IOS操作系统或者MAC OS操作系统，也可以是谷歌(google)公司的安卓(Android)操作系统，也可以是Unix操作系统，也可以是Linux操作系统，也可以是基于上述系统的再开发的改进操作系统；同时，在本实施例中，对操作系统的版本也不做具体限定。

[0147] 在本实施例中，操作系统的标识包括以下标识中的一种或多种的组合：操作系统的类型标识，操作系统的名称标识及操作系统的版本标识等；操作系统的配置信息包括以下配置信息中的一种或多种的组合：操作系统中与该任一应用程序中功能组件相关的功能组件配置信息，以及操作系统中与该任一应用程序中功能组件相关的硬件模块配置信息。

[0148] 在本实施例中，该任一应用程序的标识包括以下标识中的一种或多种的组合：应用程序的类型标识，应用程序的版本标识，及应用程序的包标识等；应用程序的配置信息包括：应用程序的功能组件配置信息等。

[0149] 通过对本计算机设备中应用程序中代码集在内存中的加载状态实时监控，并实时将包含应用程序中代码集在内存中的加载状态实时监控信息的内存管理配置信息上传至服务器，以供服务器将内存管理配置信息共享至其他计算机设备，有效提高了其他计算机设备的内存管理精准度，有效保证了其他计算机设备的运行速度，并有效降低了其他计算机设备的运行电量。

[0150] 可选地，在监测到所述任一应用程序的版本已更新的情况下，该内存管理方法在步骤S502之后还包括步骤S503和步骤S504；

[0151] 步骤S503，当在操作系统中运行版本已更新的该任一应用程序时，在内存中加载版本已更新的该任一应用程序已运行功能组件对应的第九代码集，并在预设的存储器中加载版本已更新的该任一应用程序未运行功能组件对应的第十代码集；

[0152] 步骤S504，将当前内存管理配置信息上传至设定服务器；其中，当前内存管理配置信息包括：操作系统的标识和配置信息，该任一应用程序的当前标识和当前配置信息，以及第九代码集的标识和第十代码集的标识。

[0153] 通过步骤S503和步骤S504，通过对本计算机设备中版本已更新的应用程序中代码集在内存中的加载状态实时监控，并实时将包含版本已更新的应用程序中代码集在内存中的加载状态实时监控信息的内存管理配置信息上传至服务器，以供服务器将内存管理配置信息共享至其他计算机设备，有效提高了其他计算机设备的内存管理精准度，有效保证了其他计算机设备的运行速度，并有效降低了其他计算机设备的运行电量。

[0154] 本发明第五实施例所述的内存管理方法，实现了对内存的精准管理，避免了在内存中加载过多的应用程序代码集，导致内存占用过多，导致计算机终端运行缓慢，以及耗电

量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0155] 本发明第六实施例，一种内存管理装置，如图9所示，包括以下组成部分：

[0156] 处理器501和存储器502。在本发明的一些实施例中，处理器50和1存储器502可通过总线或者其它方式连接。

[0157] 处理器501可以是通用处理器，例如中央处理器(Central ProcessingUnit，CPU)，还可以是数字信号处理器(Digital Signal Processor，DSP)、专用集成电路(英文：Application Specific Integrated Circuit，ASIC)，或者是被配置成实施本发明实施例的一个或多个集成电路。其中，存储器502用于存储所述处理器501的可执行指令；

[0158] 存储器502，用于存储程序代码，并将该程序代码传输给处理器501。存储器502可以包括易失性存储器(Volatile Memory)，例如随机存取存储器(Random Access Memory，RAM)；存储器502也可以包括非易失性存储器(Non-Volatile Memory)，例如只读存储器(Read-Only Memory，ROM)、快闪存储器(Flash Memory)、硬盘(Hard Disk Drive，HDD)或固态硬盘(Solid-State Drive，SSD)；存储器502还可以包括上述种类的存储器的组合。

[0159] 其中，处理器501用于调用所述存储器502存储的程序代码管理代码，执行本发明第一实施例至本发明第五实施例中任一实施例中部分或全部步骤。

[0160] 本发明第六实施例中所述的内存管理装置，实现了对内存的精准管理，避免了在内存中加载过多的应用程序代码集，导致内存占用过多，导致计算机终端运行缓慢，以及耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0161] 本发明第七实施例，一种计算机可读存储介质。

[0162] 计算机存储介质可以是RAM存储器、闪存、ROM存储器、EPROM存储器、EEPROM存储器、寄存器、硬盘、移动硬盘、CD-ROM或者本领域已知的任何其他形式的存储介质。

[0163] 计算机可读存储介质存储有一个或者多个程序，该一个或者多个程序可被一个或者多个处理器执行，以实现本发明第一实施例至本发明第五实施例中任一实施例中部分或全部步骤。

[0164] 本发明第七实施例中所述的计算机可读存储介质，存储有一个或者多个程序，该一个或者多个程序可被一个或者多个处理器执行，能够实现对内存的精准管理，避免了在内存中加载过多的应用程序代码集，导致内存占用过多，导致计算机终端运行缓慢，以及耗电量过大的弊端，有效保证了计算机设备的运行速度，并有效降低了计算机设备的运行电量。

[0165] 需要说明的是，在本文中，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

[0166] 上述本发明实施例序号仅仅为了描述，不代表实施例的优劣。

[0167] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本发明的技术方案本质上或者说对现有技术做

出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中，包括若干指令用以使得一台终端(可以是手机，计算机，服务器，空调器，或者网络设备等)执行本发明各个实施例所述的方法。

[0168] 上面结合附图对本发明的实施例进行了描述，但是本发明并不局限于上述的具体实施方式，上述的具体实施方式仅仅是示意性的，而不是限制性的，本领域的普通技术人员在本发明的启示下，在不脱离本发明宗旨和权利要求所保护的范围情况下，还可做出很多形式，这些均属于本发明的保护之内。

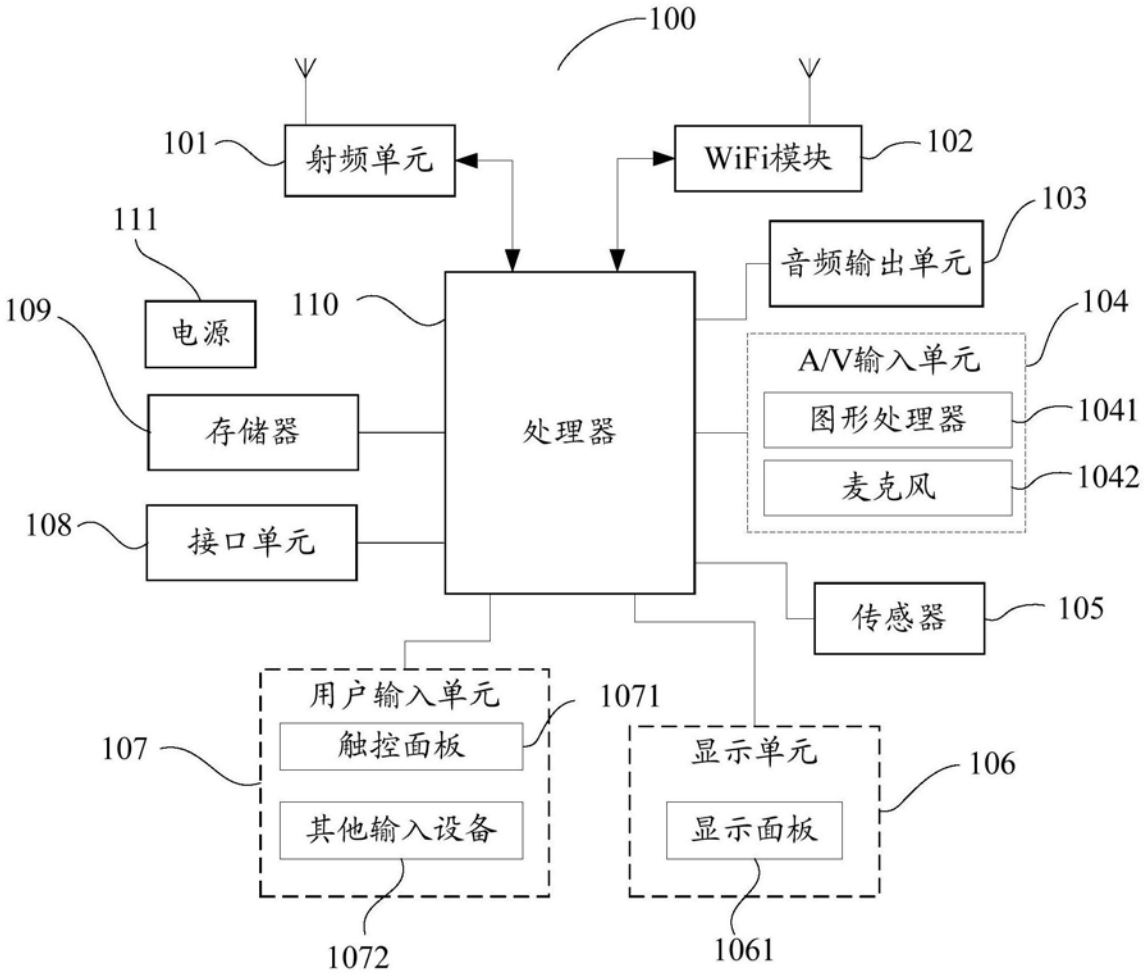

100
101
射频单元
WiFi模块
102
103
音频输出单元
111
电源
109
110
处理器
104
A/V输入单元
存储器
图形处理器
1041
麦克风
1042
108
接口单元
传感器
105
用户输入单元
1071
107
触控面板
显示单元
106
其他输入设备
显示面板
1072
1061

图1

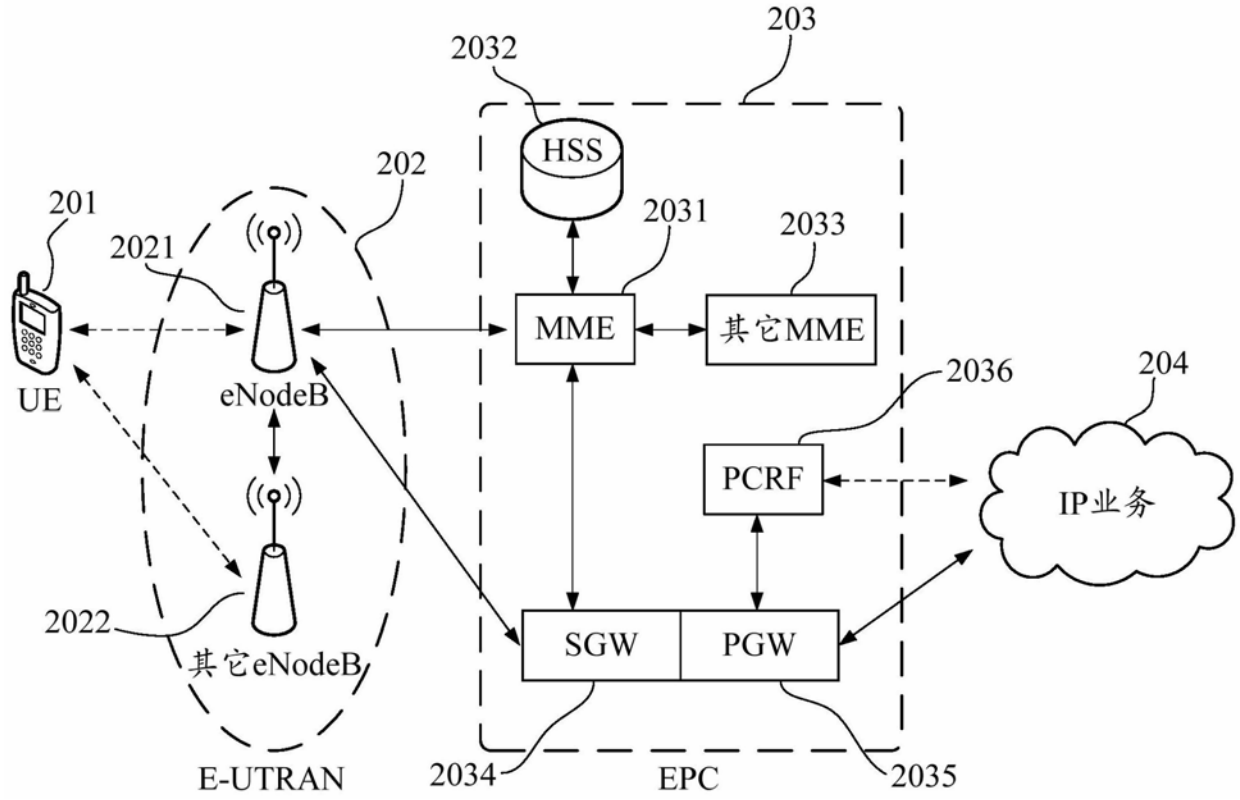

203
2032
HSS
202
201
2031
2033
2021
MME
其它MME
UE
eNodeB
2036
204
PCRF
IP业务
2022
其它eNodeB
SGW
PGW
E-UTRAN
2034
EPC
2035

图2

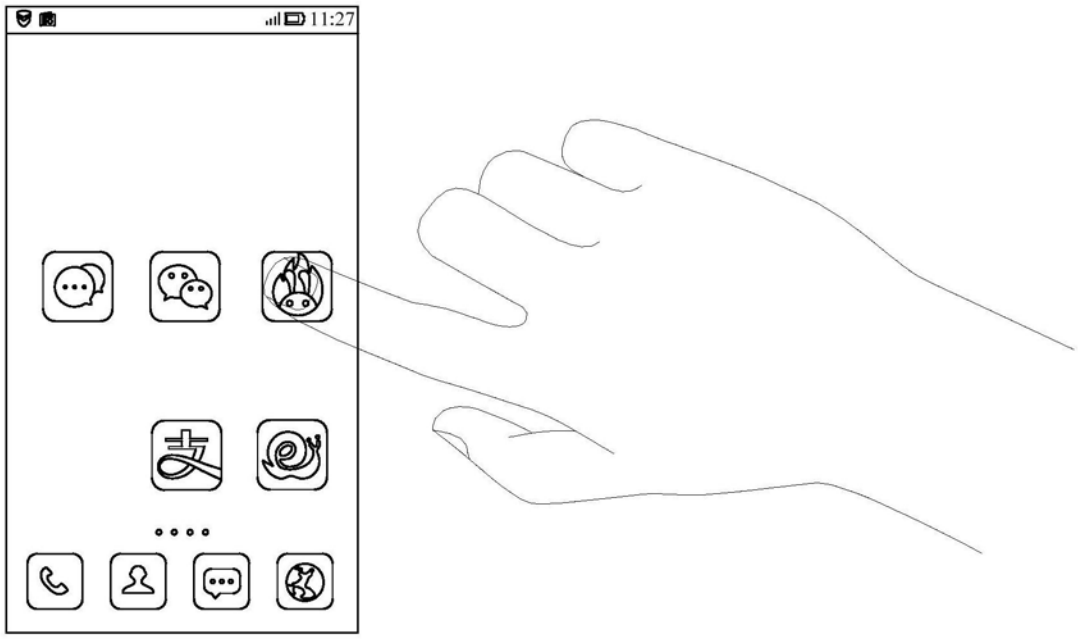

11:27

图3

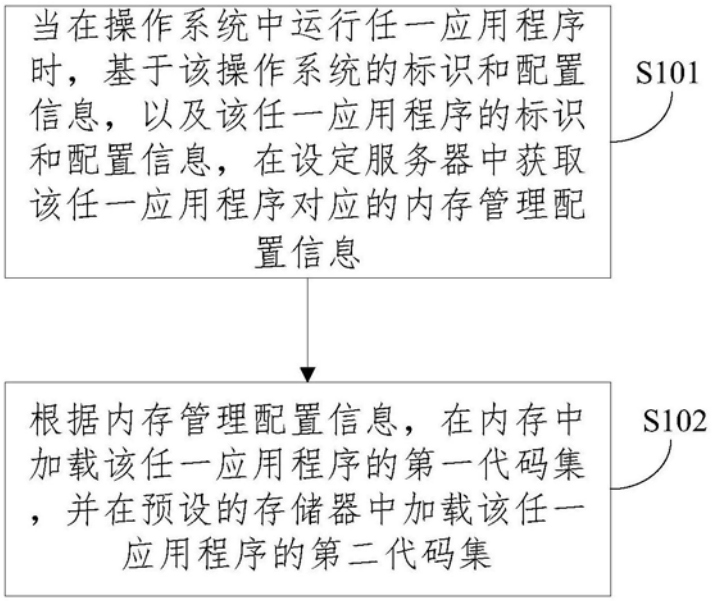

当在操作系统中运行任一应用程序时，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中获取该任一应用程序对应的内存管理配置信息
S101
根据内存管理配置信息，在内存中加载该任一应用程序的第一代码集，并在预设的存储器中加载该任一应用程序的第二代码集
S102

图4

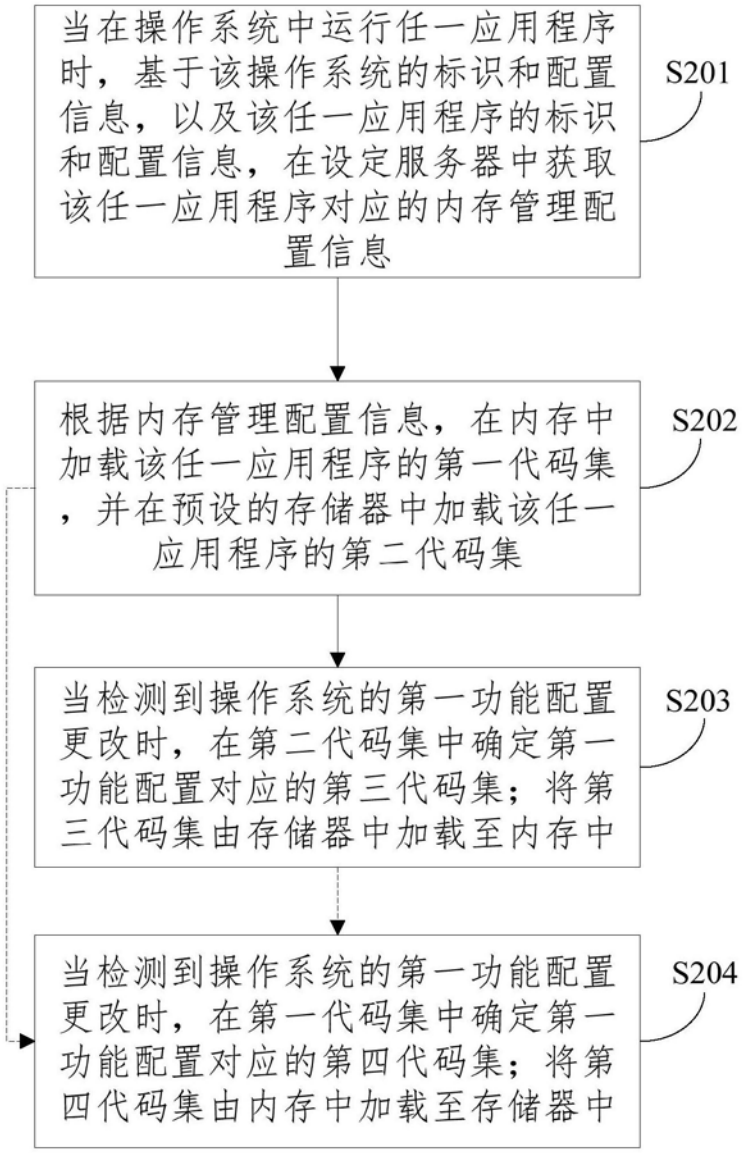

当在操作系统中运行任一应用程序时，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中获取该任一应用程序对应的内存管理配置信息
S201
根据内存管理配置信息，在内存中加载该任一应用程序的第一代码集，并在预设的存储器中加载该任一应用程序的第二代码集
S202
当检测到操作系统的第一功能配置更改时，在第二代码集中确定第一功能配置对应的第三代码集；将第三代码集由存储器中加载至内存中
S203
当检测到操作系统的第一功能配置更改时，在第一代码集中确定第一功能配置对应的第四代码集；将第四代码集由内存中加载至存储器中
S204

图5

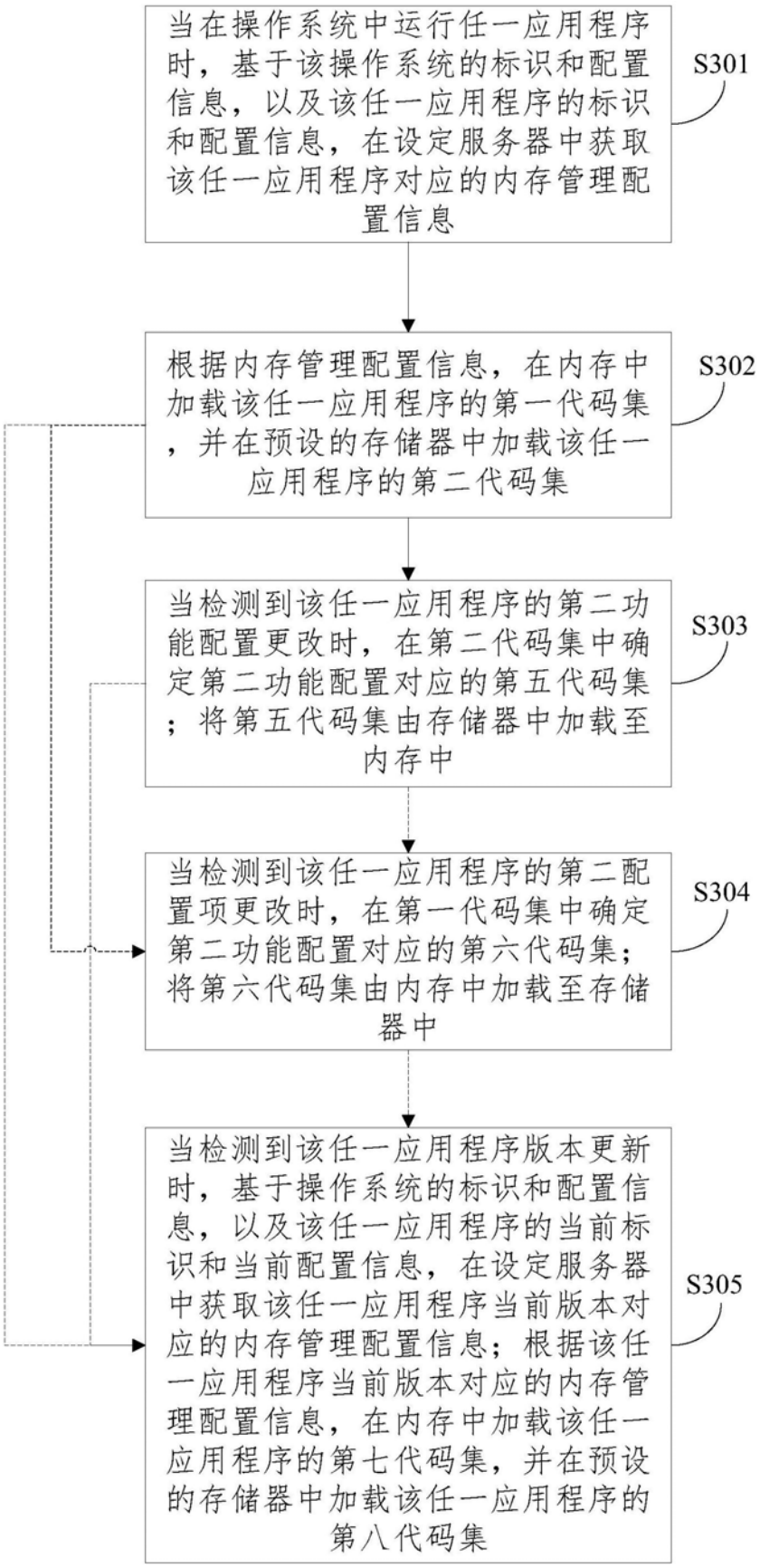

当在操作系统中运行任一应用程序时，基于该操作系统的标识和配置信息，以及该任一应用程序的标识和配置信息，在设定服务器中获取该任一应用程序对应的内存管理配置信息
S301
根据内存管理配置信息，在内存中加载该任一应用程序的第一代码集，并在预设的存储器中加载该任一应用程序的第二代码集
S302
当检测到该任一应用程序的第二功能配置更改时，在第二代码集中确定第二功能配置对应的第五代码集；将第五代码集由存储器中加载至内存中
S303
当检测到该任一应用程序的第二配置项更改时，在第一代码集中确定第二功能配置对应的第六代码集；将第六代码集由内存中加载至存储器中
S304
当检测到该任一应用程序版本更新时，基于操作系统的标识和配置信息，以及该任一应用程序的当前标识和当前配置信息，在设定服务器中获取该任一应用程序当前版本对应的内存管理配置信息；根据该任一应用程序当前版本对应的内存管理配置信息，在内存中加载该任一应用程序的第七代码集，并在预设的存储器中加载该任一应用程序的第八代码集
S305

图6

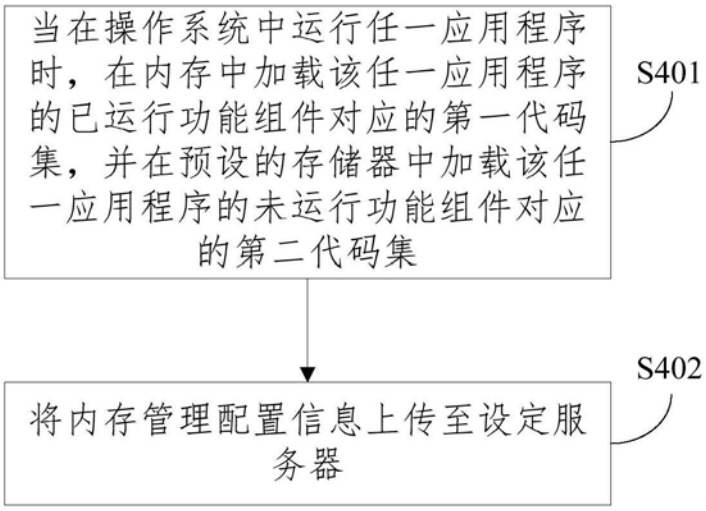

当在操作系统中运行任一应用程序时，在内存中加载该任一应用程序的已运行功能组件对应的第一代码集，并在预设的存储器中加载该任一应用程序的未运行功能组件对应的第二代码集
S401
将内存管理配置信息上传至设定服务器
S402

图7

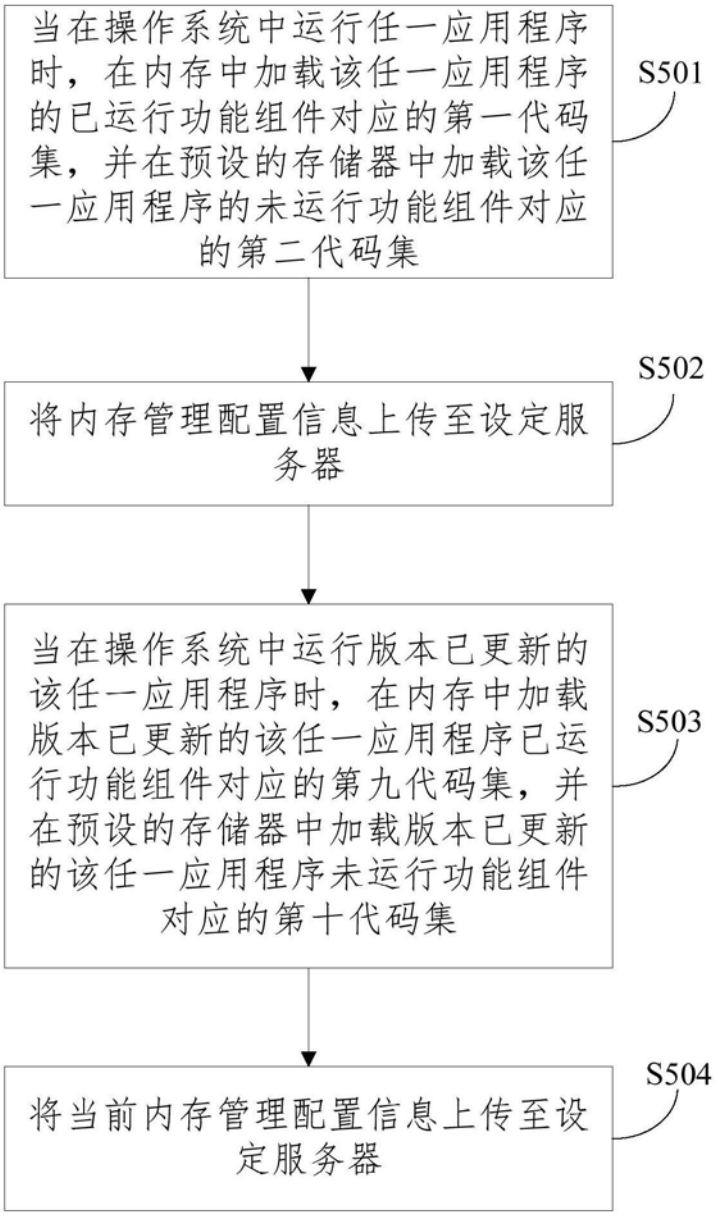

当在操作系统中运行任一应用程序时，在内存中加载该任一应用程序的已运行功能组件对应的第一代码集，并在预设的存储器中加载该任一应用程序的未运行功能组件对应的第二代码集
S501
将内存管理配置信息上传至设定服务器
S502
当在操作系统中运行版本已更新的该任一应用程序时，在内存中加载版本已更新的该任一应用程序已运行功能组件对应的第九代码集，并在预设的存储器中加载版本已更新的该任一应用程序未运行功能组件对应的第十代码集
S503
将当前内存管理配置信息上传至设定服务器
S504

图8

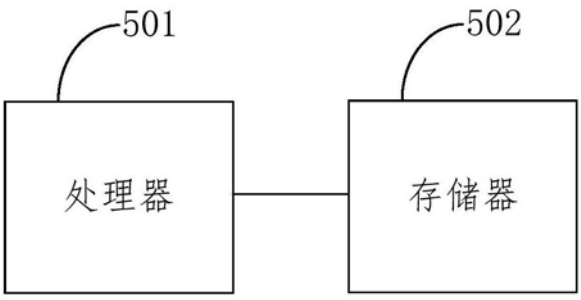

501
502
处理器
存储器

图9