



(12)发明专利

(10)授权公告号 CN 106708543 B

(45)授权公告日 2019.12.13

(21)申请号 201510438399.6

(22)申请日 2015.07.22

(65)同一申请的已公布的文献号
申请公布号 CN 106708543 A

(43)申请公布日 2017.05.24

(73)专利权人 TCL集团股份有限公司
地址 516006 广东省惠州市仲恺高新技术
开发区十九号小区

(72)发明人 彭芑

(74)专利代理机构 深圳中一专利商标事务所
44237

代理人 张全文

(51)Int.Cl.
G06F 8/65(2018.01)

(56)对比文件

CN 103176824 A,2013.06.26,
CN 103019775 A,2013.04.03,
CN 103455348 A,2013.12.18,
US 2014317614 A1,2014.10.23,

审查员 成晓潇

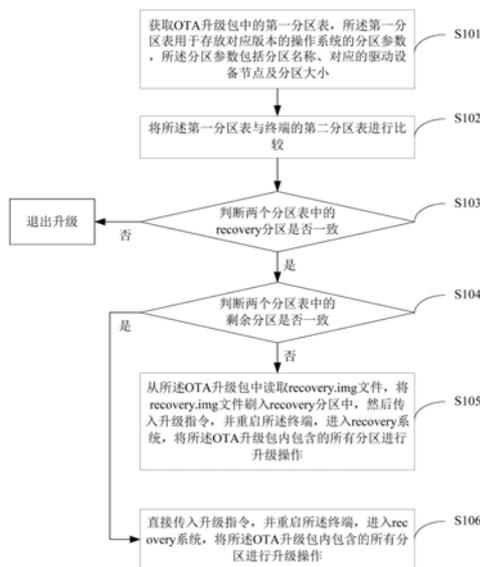
权利要求书2页 说明书6页 附图2页

(54)发明名称

一种操作系统的OTA升级方法及装置

(57)摘要

本发明适用于计算机技术领域,提供了一种操作系统的OTA升级方法及装置,包括:获取OTA升级包中的第一分区表;将所述第一分区表与终端的第二分区表进行比较;判断两个分区表中的recovery分区是否一致;若一致,判断两个分区表中的剩余分区是否一致;若所述两个分区表中的剩余分区的比较结果不一致,则从所述OTA升级包中读取recovery.img文件,将recovery.img文件刷入recovery分区中;传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作。本发明使得在升级后的操作系统对系统分区有调整时,仍然能够利用OTA完成操作系统升级,避免了分区调整对操作系统升级所带来的影响。



1. 一种操作系统的空间下载技术OTA升级方法,其特征在于,包括:

获取OTA升级包中的第一分区表,所述第一分区表用于存放所述OTA升级包对应版本的操作系统的分区参数,所述分区参数包括分区名称、对应的驱动设备节点及分区大小;

将所述第一分区表与终端的第二分区表进行比较,所述第二分区表,其存储的是所述终端当前版本的操作系统的分区参数;

判断两个分区表中的recovery分区是否一致;

若两个分区表中的recovery分区不一致,则退出升级;若两个分区表中的recovery分区一致,则判断两个分区表中的剩余分区是否一致;

若所述两个分区表中的剩余分区的比较结果不一致,则从所述OTA升级包中读取recovery.img文件,将recovery.img文件刷入recovery分区中,然后传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作;

若所述两个分区表中的剩余分区的比较结果一致,则直接传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作。

2. 如权利要求1所述的方法,其特征在于,所述方法还包括:

将所述第一分区表的数据更新至所述第二分区表中。

3. 如权利要求2所述的方法,其特征在于,所述将所述第一分区表的数据更新至所述第二分区表中包括:

将所述第一分区表写入所述终端更新后的操作系统的系统分区的指定路径,以作为更新后的所述第二分区表存储。

4. 如权利要求1所述的方法,其特征在于,在所述获取OTA升级包中的第一分区表之前,所述方法还包括:

设定所述第一分区表中recovery分区的驱动设备节点及分区大小始终固定不变。

5. 如权利要求4所述的方法,其特征在于,所述设定所述第一分区表中recovery分区的驱动设备节点及分区大小始终固定不变包括:

设定所述第一分区表中recovery分区为所述OTA升级包对应版本的操作系统的第一个分区,且设定该recovery分区的分区大小固定不变。

6. 一种操作系统的空间下载技术OTA升级装置,其特征在于,包括:

获取单元,用于获取OTA升级包中的第一分区表,所述第一分区表用于存放对应版本的操作系统的分区参数,所述分区参数包括分区名称、对应的驱动设备节点及分区大小;

比较单元,用于将所述第一分区表与终端的第二分区表进行比较,所述第二分区表,其存储的是所述终端当前版本的操作系统的分区参数;

判断单元,用于判断两个分区表中的recovery分区是否一致;

处理单元,用于若两个分区表中的recovery分区不一致,则退出升级;若两个分区表中的recovery分区一致,则判断两个分区表中的剩余分区是否一致;

第一升级单元,用于若所述两个分区表中的剩余分区的比较结果不一致,则从所述OTA升级包中读取recovery.img文件,将recovery.img文件刷入recovery分区中,然后传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作;

第二升级单元,用于若所述两个分区表中的剩余分区的比较结果一致,则直接传入升

级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作。

7.如权利要求6所述的装置,其特征在于,所述装置还包括:

更新单元,用于将所述第一分区表的数据更新至所述第二分区表中。

8.如权利要求7所述的装置,其特征在于,所述更新单元具体用于:

将所述第一分区表写入所述终端更新后的操作系统的系统分区的指定路径,以作为更新后的所述第二分区表存储。

9.如权利要求6所述的装置,其特征在于,所述装置还包括:

设定单元,用于设定所述第一分区表中recovery分区的驱动设备节点及分区大小始终固定不变。

10.如权利要求9所述的装置,其特征在于,所述设定单元具体用于:

设定所述第一分区表中recovery分区为所述OTA升级包对应版本的操作系统的第一个分区,且设定该recovery分区的分区大小固定不变。

一种操作系统的OTA升级方法及装置

技术领域

[0001] 本发明属于计算机技术领域,尤其涉及一种操作系统的空间下载技术(Over-the-Air Technology,OTA)升级方法及装置。

背景技术

[0002] 对Android系统来说,其对应着多个系统分区,包括system、boot、recovery、cache、data等原生分区,同时还存在着一些自定义的扩展分区,这些不同分区有的用来存储操作系统程序,有的用来存储用户数据,有的用来实现操作系统的某些特殊功能。目前,随着应用体积的增大,可能出现某个分区的可用空间已无法允许安装新的应用,需要增大分区的情况。例如一个EMMC (Embedded Multi Media Card) 芯片的存储空间被划分为N段,每一段对应一个分区,若需要将分区2从8M扩展到10M,那么分区1无需变化,分区3的位置肯定要发生改变,而其他分区的位置也有可能发生改变,至少要保证从其他分区中减少2M的存储空间,以划分给分区2,才能保证芯片的整体存储空间大小不变。

[0003] 目前,若要对Android系统进行OTA升级,其前提是新版本系统的系统分区(包括分区的名称、大小、节点等信息)无修改,但如前文所述,在实际的操作系统版本升级过程中,会出现对现有分区进行修改、调整的情况,现有的OTA升级方法则无法适用于该情况,导致Android系统的OTA升级不成功。

发明内容

[0004] 本发明实施例的目的在于提供一种操作系统的OTA升级方法及装置,旨在解决现有的操作系统OTA升级方法无法适用于系统分区已修改的情况,导致OTA升级不成功的问题。

[0005] 本发明实施例是这样实现的,一种操作系统的空间下载技术OTA升级方法,包括:

[0006] 获取OTA升级包中的第一分区表,所述第一分区表用于存放对应版本的操作系统的分区参数,所述分区参数包括分区名称、对应的驱动设备节点及分区大小;

[0007] 将所述第一分区表与终端的第二分区表进行比较;

[0008] 判断两个分区表中的recovery分区是否一致;若两个分区表中的recovery分区不一致,则退出升级;若两个分区表中的recovery分区一致,则判断两个分区表中的剩余分区是否一致;

[0009] 若所述两个分区表中的剩余分区的比较结果不一致,则从所述OTA升级包中读取recovery.img文件,将recovery.img文件刷入recovery分区中,然后传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作;

[0010] 若所述两个分区表中的剩余分区的比较结果一致,则直接传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作。

[0011] 本发明实施例的另一目的在于提供一种操作系统的空间下载技术OTA升级装置,包括:

[0012] 获取单元,用于获取OTA升级包中的第一分区表,所述第一分区表用于存放对应版本的操作系统的分区参数,所述分区参数包括分区名称、对应的驱动设备节点及分区大小;

[0013] 比较单元,用于将所述第一分区表与终端的第二分区表进行比较;

[0014] 判断单元,用于判断两个分区表中的recovery分区是否一致;

[0015] 处理单元,用于若两个分区表中的recovery分区不一致,则退出升级;若两个分区表中的recovery分区一致,则判断两个分区表中的剩余分区是否一致;

[0016] 第一升级单元,用于若所述两个分区表中的剩余分区的比较结果不一致,则从所述OTA升级包中读取recovery.img文件,将recovery.img文件刷入recovery分区中,然后传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作;

[0017] 第二升级单元,用于若所述两个分区表中的剩余分区的比较结果一致,则直接传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作。

[0018] 通过本发明实施例所示方案,使得在升级后的操作系统对系统分区有调整时,仍然能够利用OTA完成操作系统升级,避免了分区调整对操作系统升级过程所带来的影响,保证了操作系统的升级效率。

附图说明

[0019] 为了更清楚地说明本发明实施例中的技术方案,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,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。

[0020] 图1是本发明实施例一提供的操作系统的OTA升级方法的实现流程图;

[0021] 图2是本发明实施例四提供的操作系统的OTA升级装置的结构框图。

具体实施方式

[0022] 以下描述中,为了说明而不是为了限定,提出了诸如特定系统结构、技术之类的具体细节,以便透切理解本发明实施例。然而,本领域的技术人员应当清楚,在没有这些具体细节的其它实施例中也可以实现本发明。在其它情况中,省略对众所周知的系统、装置、电路以及方法的详细说明,以免不必要的细节妨碍本发明的描述。

[0023] 在本发明实施例中,所述操作系统,可以为适用于本方案的移动终端的智能操作系统,包括但不限于Android、iOS、Symbian、Windows Phone和BlackBerry OS等,以下实施例仅以Android系统为例进行说明,其他智能操作系统只要能够支持本方案实现的,均可以按相同或相似原理实现本方案,在此不一一赘述。

[0024] 为了说明本发明所述的技术方案,下面通过具体实施例来进行说明。

[0025] 实施例一:

[0026] 图1示出了本发明实施例一提供的操作系统的OTA升级方法的实现流程,详述如下:

[0027] 在步骤S101中,获取OTA升级包中的第一分区表,所述第一分区表用于存放对应版

本的操作系统的分区参数,所述分区参数包括分区名称、对应的驱动设备节点及分区大小。

[0028] OTA升级包由操作系统的升级服务器提供,需要进行系统升级的终端设备可以通过升级应用从升级服务器中下载OTA升级包到本地。所述分区表专门用于存储其对应版本的操作系统的分区参数,其中,分区参数至少包括了分区名称、对应的驱动设备节点及分区大小,例如:

[0029] /recovery/dev/block/mmcblk0p12000000,代表Android系统中的recovery分区对应的驱动设备节点是/dev/block/mmcblk0p1,其为第1个分区,对应的分区大小是2000000字节;

[0030] /system/dev/block/mmcblk0p3800000000,代表Android系统中的system分区对应的驱动设备节点是/dev/block/mmcblk0p3,其为第3个分区,对应的分区大小是8000000000字节。

[0031] 在本实施例中,对于第一分区表来说,其存储的是OTA升级包对应的升级后的操作系统的分区参数。

[0032] 在步骤S102中,将所述第一分区表与终端的第二分区表进行比较。

[0033] 所述第二分区表,其存储的是终端设备当前版本的操作系统的分区参数。

[0034] 通过将第一分区表与第二分区表进行比较,若此次系统升级未对系统分区进行调整,那么第一分区表与第二分区表的数据是一致的,若第一分区表与第二分区表的比较结果不一致,则表明此次系统升级对系统分区进行了调整。

[0035] 在步骤S103中,判断两个分区表中的recovery分区是否一致。

[0036] 在步骤S104中,若两个分区表中的recovery分区不一致,则退出升级;若两个分区表中的recovery分区一致,则判断两个分区表中的剩余分区是否一致。

[0037] 基于步骤S102的比较结果,首先,判断两个分区表中的recovery分区是否一致。

[0038] 由于本发明实施例后续升级方案能够成功进行的前提,在于操作系统的recovery分区在升级前后未修改(包括recovery分区的分区名称、大小、节点等分区信息均未发生改变),因此,若判断出第一分区表与第二分区表中关于recovery分区的分区信息不一致,后续升级方案则无法成功进行,那么,直接终止并退出升级过程。若第一分区表与第二分区表中关于recovery分区的分区信息一致,则继续执行后续方案,检测两个分区表中除recovery分区之外的剩余分区是否一致。

[0039] 在步骤S105中,若所述两个分区表中的剩余分区的比较结果不一致,则从所述OTA升级包中读取recovery.img文件,将recovery.img文件刷入recovery分区中,然后传入升级指令,并重启所述终端,进入recovery系统,将OTA升级包内包含的所有分区进行升级操作。

[0040] 在S106中,若所述两个分区表中的剩余分区的比较结果一致,则直接传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作。

[0041] 一旦recovery.img文件刷入recovery分区中,则可以传入升级指令并重启终端,然后进入到recovery系统中去,recovery系统的kernel层根据新的分区情况去生成新的驱动设备节点,运行升级程序,将OTA升级包内的数据依次写入到新划分的各个分区中,并在

完成之后重启进入操作系统,从而完成操作系统的升级过程。

[0042] 通过本发明实施例一所示方案,使得在升级后的操作系统对系统分区有调整时,仍然能够利用OTA完成操作系统升级,避免了分区调整对操作系统升级过程所带来的影响,保证了操作系统的升级效率。

[0043] 实施例二:

[0044] 本实施例与本发明实施例一大致相同,区别在于步骤S105进一步包括:将所述第一分区表的数据更新至所述第二分区表中。

[0045] 在本实施例中,当完成系统升级过程之后,对终端设备本地存储的第二分区表进行更新,以保证当下一次进行操作系统升级时,可以将终端设备本地实际的分区情况与升级版本的操作系统的分区情况进行比较,保证分区情况比较结果的准确性。

[0046] 具体地,可以通过如下方式实现:

[0047] 在制作升级包时,将第一分区表存放在系统分区(以下简称system分区)的指定路径中,例如,存放在ststem/etc目录之下,那么在升级过程中,升级包内的数据会依次写入到各个分区,则第一分区表就会被存储在终端设备本地对应的指定路径中,从而完成本地存放的第二分区表的更新过程。

[0048] 实施例三:

[0049] 本实施例在本发明实施例一的基础上,增加了对recovery分区相关的分区参数的限定步骤,在步骤S101之前进一步包括以下步骤:设定所述第一分区表中recovery分区的驱动设备节点及分区大小始终固定不变。

[0050] 在本发明前述实施例中,在对第一分区表和第二分区表的比较过程中,首先可以比较二者关于recovery分区的相关分区参数是否发生了变化,若recovery分区也发生了变化,则OTA升级方法也不再适用,会直接退出升级过程。在本实施例中,可以设定无论操作系统如何升级,recovery分区相关的分区参数始终固定不变,以保证OTA升级方法的顺利执行。

[0051] 将recovery分区始终设置为操作系统的第一个分区且分区大小不变,这样无论实际的系统分区需求如何变化,升级过程中对其他分区怎样调整,都不会影响到recovery分区的分区参数。且由于recovery分区是专门用于操作系统升级的最小系统,功能单一,自身有变化的可能性也较小,因此,通过本实施例的方案限制,并不会对整个操作系统升级方案产生较大的影响。

[0052] 实施例四:

[0053] 对应于上文实施例所述的操作系统的OTA升级方法,图2示出了本发明操作系统的OTA升级装置的结构框图,所述操作系统的OTA升级装置可以是内置于智能终端设备或智能终端设备的操作系统内的软件单元、硬件单元或者是软硬结合的单元。为了便于说明,仅示出了与本实施例相关的部分。

[0054] 参照图2,该装置包括:

[0055] 获取单元21,获取OTA升级包中的第一分区表,所述第一分区表用于存放对应版本的操作系统的分区参数,所述分区参数包括分区名称、对应的驱动设备节点及分区大小;

[0056] 比较单元22,将所述第一分区表与终端的第二分区表进行比较;

[0057] 判断单元23,判断两个分区表中的recovery分区是否一致;

[0058] 处理单元24,若两个分区表中的recovery分区不一致,则退出升级;若两个分区表中的recovery分区一致,则判断两个分区表中的剩余分区是否一致;

[0059] 第一升级单元25,若所述两个分区表中的剩余分区的比较结果不一致,则从所述OTA升级包中读取recovery.img文件,将recovery.img文件刷入recovery分区中,然后传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作;

[0060] 第二升级单元26,若所述两个分区表中的剩余分区的比较结果一致,则直接传入升级指令,并重启所述终端,进入recovery系统,将所述OTA升级包内包含的所有分区进行升级操作。

[0061] 可选地,所述装置还包括:

[0062] 更新单元,将所述第一分区表的数据更新至所述第二分区表中。

[0063] 可选地,所述更新单元具体用于:

[0064] 将所述第一分区表写入所述操作系统的系统分区的指定路径,以作为更新后的所述第二分区表存储。

[0065] 可选地,所述装置还包括:

[0066] 设定单元,设定所述第一分区表中recovery分区的驱动设备节点及分区大小始终固定不变。

[0067] 可选地,所述设定单元具体用于:

[0068] 设定所述第一分区表中recovery分区为所述操作系统的第一个分区,且设定该recovery分区的分区大小固定不变。

[0069] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,仅以上述各功能单元、模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能单元、模块完成,即将所述装置的内部结构划分成不同的功能单元或模块,以完成以上描述的全部或者部分功能。实施例中的各功能单元、模块可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中,上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。另外,各功能单元、模块的具体名称也只是为了便于相互区分,并不用于限制本申请的保护范围。上述系统中单元、模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0070] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0071] 在本发明所提供的实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的系统实施例仅仅是示意性的,例如,所述模块或单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通讯连接可以是通过一些接口,装置或单元的间接耦合或通讯连接,可以是电性,机械或其它的形式。

[0072] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0073] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0074] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明实施例的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或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中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)或处理器(processor)执行本发明实施例各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0075] 以上所述实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明实施例各实施例技术方案的精神和范围。

[0076] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

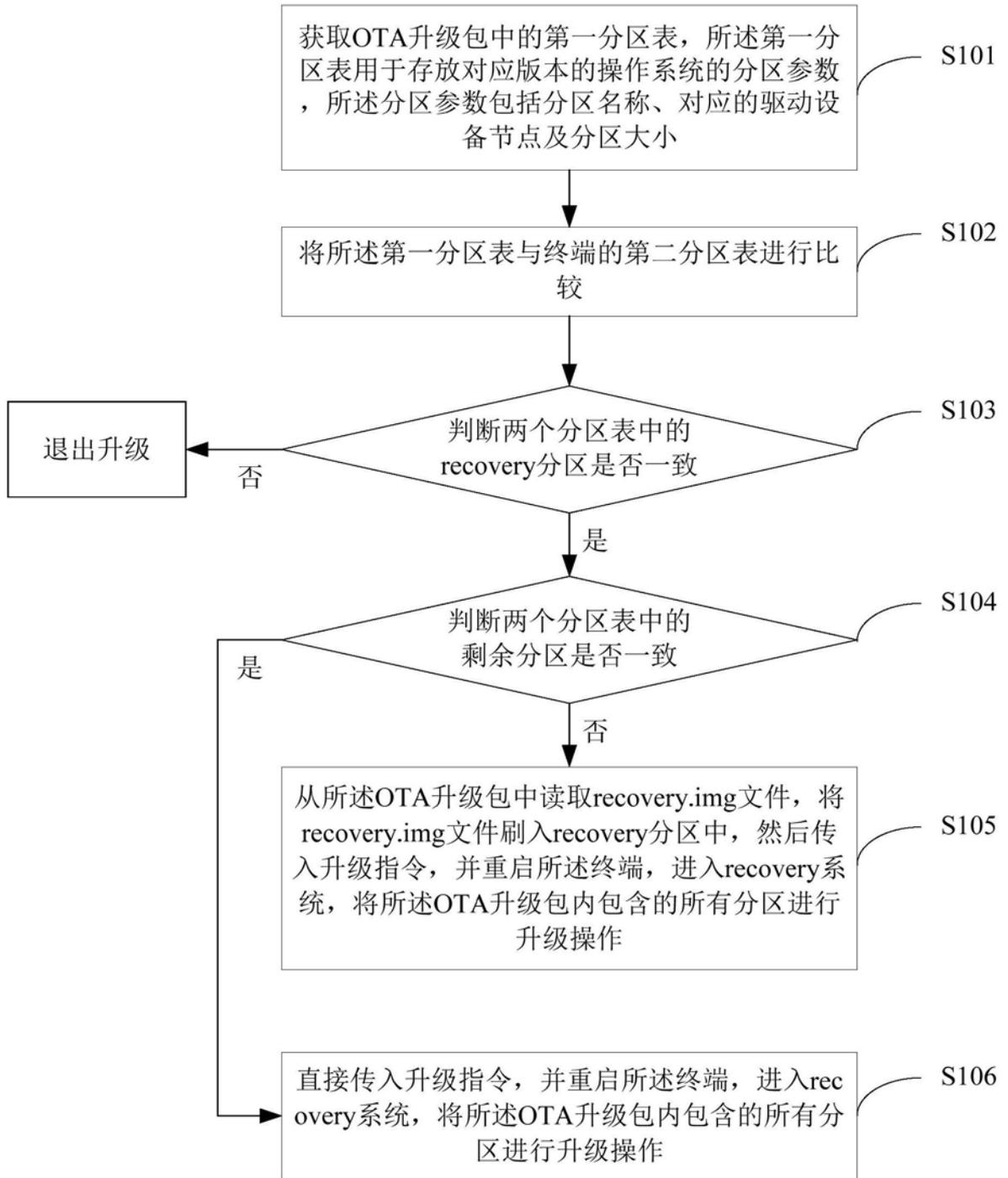


图1

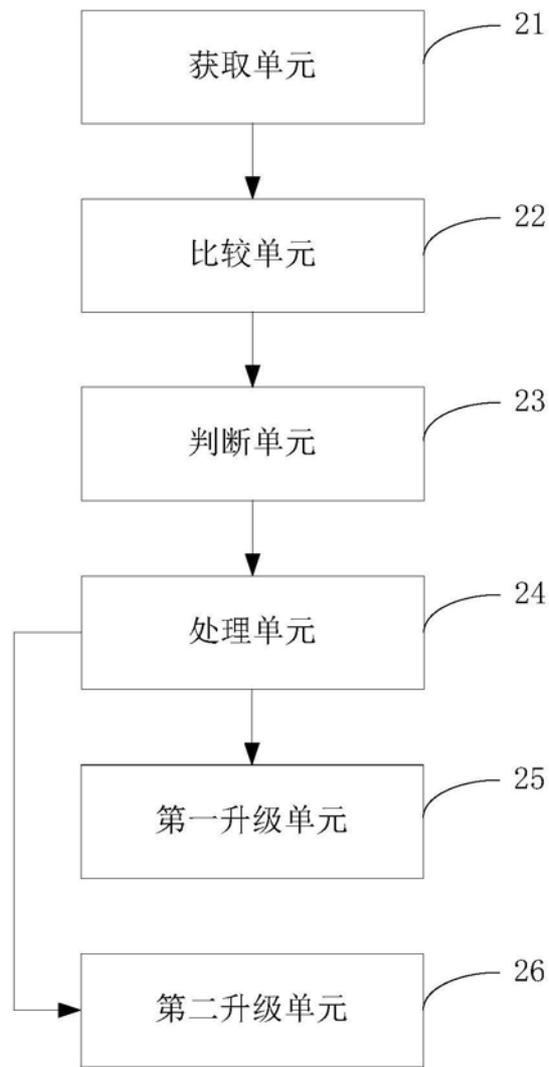


图2