



(12) 发明专利

(10) 授权公告号 CN 102298538 B

(45) 授权公告日 2014. 03. 26

(21) 申请号 201010570595. 6

代理人 马浩

(22) 申请日 2010. 09. 25

(51) Int. Cl.

(30) 优先权数据

G06F 9/48 (2006. 01)

61/321, 616 2010. 04. 07 US

(56) 对比文件

(73) 专利权人 苹果公司

CN 101667208 A, 2010. 03. 10, 全文.

地址 美国加利福尼亚

CN 101667208 A, 2010. 03. 10, 说明书 1 - 2

页, 图 1 - 2.

(72) 发明人 E · 阿尔伯特 L · 巴拉德

审查员 刘娜

G · R · 查普曼 N · G · 克兰

小 T · B · 达菲 S · 福斯塔

G · J · 弗里德曼 D · W · 古德温

J · 亚罗西 D · 利津格

C · 玛塞林诺 H · G · 马森

C · 玛斯普特 D · 梅斯泽维斯基

G · 诺威克 C · 罗赛特 R · 施赖尔

小 J · 索科尔 D · 索里索

C · 斯苏瓦纳努科恩 M · 范米利甘

M · G · 沃特森

(74) 专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

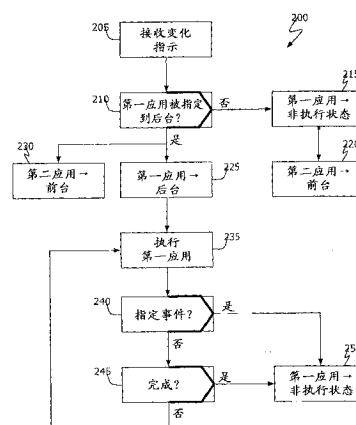
权利要求书3页 说明书10页 附图6页

(54) 发明名称

机会性的多任务处理

(57) 摘要

本公开涉及机会性的多任务处理。提供用于个人电子设备的服务,通过这些服务支持后台处理或多任务处理的形式。所公开的服务允许用户应用利用后台处理而对前台进程的用户体验或个人电子设备的功率资源没有显著负面结果。为了实现所公开的多任务处理,许多操作约束中的一个或多个可以被强制实施。这些约束的一个结果可能是,进程将不能在后台进行其如果在前台则可能能够做的工作。所公开的服务的实施对执行中的用户应用可以是基本上透明的,并且在某些情况下,可以无需用户应用的明确协作而执行。



1. 一种多任务处理方法,包括:
识别以前台状态执行的第一用户应用;
接收指示第二用户应用将被置于前台状态的用户输入;
只有当第一用户应用已经将自身识别为能够以后台状态执行的用户应用时,才响应于所接收的用户输入将第一用户应用置于后台状态;和
将第二用户应用置于前台状态。
2. 根据权利要求1的方法,进一步包括:如果第一用户应用没有将自身识别为能够以后台状态执行的用户应用,则将第一用户应用置于非执行状态。
3. 根据权利要求1的方法,进一步包括:如果第一用户应用试图访问指定的硬件资源,则在第一用户应用被置于后台状态后,将第一用户应用置于非执行状态。
4. 根据权利要求1的方法,进一步包括:在第一用户进程完成任务之后,将第一用户应用转换为挂起状态,所述任务是第一用户应用在被置于后台状态时执行的任务。
5. 根据权利要求1的方法,进一步包括:在第一用户进程在后台状态中已经执行了指定的时间量之后,将第一用户应用转换为挂起状态。
6. 根据权利要求1的方法,进一步包括:如果第一用户应用获得共享系统资源上的锁,则将第一用户应用从后台状态转换为终止状态。
7. 一种多任务处理方法,包括:
识别以前台状态执行音频操作的第一用户应用;
接收确定第二用户应用将被置于前台状态的的第一用户输入;
响应于接收的第一用户输入,将第一用户应用置于后台状态,其中第一用户应用继续执行音频操作;
将第二用户应用置于该前台状态;
接收指示第一用户应用应当被暂停的第二用户输入;和
响应于接收的第二用户输入,将第一用户应用置于非执行状态。
8. 根据权利要求7的方法,进一步包括:
接收指示第一用户应用应当继续的第三用户输入;和
将第一用户应用从非执行状态转换为后台状态,以继续执行音频操作。
9. 一种多任务处理方法,包括:
接收针对挂起的网络通信应用的消息;
识别指定的网络通信应用;
将所识别的网络通信应用转换为后台状态;和
将所述消息提供给在后台的所识别的网络通信应用。
10. 根据权利要求9的方法,其中所述网络通信应用包括互联网协议电话 VOIP 应用。
11. 根据权利要求9的方法,进一步包括:在所述指定的网络通信应用响应所述消息后,将所述指定的网络通信应用从后台状态转换为挂起状态。
12. 一种多任务处理方法,包括:
在第一时间,发起第一用户互联网协议电话 VOIP 应用;
针对第一用户 VOIP 应用实例化一个或多个通信套接字;
在第二时间,将第一用户 VOIP 应用置于非执行状态;

在第三时间,接收与所述一个或多个通信套接字相关联的通知;
响应于接收的通知,将第一用户 VOIP 应用置于后台状态;和
在第一用户 VOIP 应用响应了该通知后,使第一用户 VOIP 应用返回到非执行状态,
其中,第一时间先于第二时间,且第二时间先于第三时间。

13. 一种多任务处理设备,包括:

用于识别以前台状态执行的第一用户应用的装置;
用于接收指示第二用户应用将被置于前台状态的用户输入的装置;
用于只有当第一用户应用已经将自身识别为能够以后台状态执行的用户应用时,才响应于所接收的用户输入将第一用户应用置于后台状态的装置;和
用于将第二用户应用置于前台状态的装置。

14. 根据权利要求 13 的设备,进一步包括:用于如果第一用户应用没有将自身识别为能够以后台状态执行的用户应用,则将第一用户应用置于非执行状态的装置。

15. 根据权利要求 13 的设备,进一步包括:用于如果第一用户应用试图访问指定的硬件资源,则在第一用户应用被置于后台状态后,将第一用户应用置于非执行状态的装置。

16. 根据权利要求 13 的设备,进一步包括:用于在第一用户进程完成任务之后,将第一用户应用转换为挂起状态的装置,所述任务是第一用户应用在被置于后台状态时执行的任务。

17. 根据权利要求 13 的设备,进一步包括:用于在第一用户进程在后台状态中已经执行了指定的时间量之后,将第一用户应用转换为挂起状态的装置。

18. 根据权利要求 13 的设备,进一步包括:用于如果第一用户应用获得共享系统资源上的锁,则将第一用户应用从后台状态转换为终止状态的装置。

19. 一种多任务处理设备,包括:

用于识别以前台状态执行音频操作的第一用户应用的装置;
用于接收确定第二用户应用将被置于前台状态的的第一用户输入的装置;
用于响应于接收的第一用户输入,将第一用户应用置于后台状态的装置,其中第一用户应用继续执行音频操作;
用于将第二用户应用置于该前台状态的装置;
用于接收指示第一用户应用应当被暂停的第二用户输入的装置;和
用于响应于接收的第二用户输入,将第一用户应用置于非执行状态的装置。

20. 根据权利要求 19 的设备,进一步包括:

用于接收指示第一用户应用应当继续的第三用户输入的装置;
用于将第一用户应用从非执行状态转换为后台状态,以继续执行音频操作的装置。

21. 一种多任务处理设备,包括:

用于接收针对挂起的网络通信应用的消息的装置;
用于识别指定的网络通信应用的装置;
用于将所识别的网络通信应用转换为后台状态的装置;和
用于将所述消息提供给在后台的所识别的网络通信应用的装置。

22. 根据权利要求 21 的设备,其中所述网络通信应用包括互联网协议电话 VOIP 应用。

23. 根据权利要求 21 的设备,进一步包括:用于在所述指定的网络通信应用响应所述

消息后,将所述指定的网络通信应用从后台状态转换为挂起状态的装置。

24. 一种多任务处理设备,包括:

用于在第一时间,发起第一用户互联网协议电话 VOIP 应用的装置;

用于针对第一用户 VOIP 应用实例化一个或多个通信套接字的装置;

用于在第二时间,将第一用户 VOIP 应用置于非执行状态的装置;

用于在第三时间,接收与所述一个或多个通信套接字相关联的通知的装置;

用于响应于接收的通知,将第一用户 VOIP 应用置于后台状态的装置;和

用于在第一用户 VOIP 应用响应了该通知后,使第一用户 VOIP 应用返回到非执行状态的装置,

其中,第一时间先于第二时间,且第二时间先于第三时间。

机会性的多任务处理

[0001] 当 Apple 的 iPhone 4 的样机在 2010 年 3 月 25 日从 Apple 的一个工程师处被窃取时,本申请中要揭示和主张的发明被提前且未经 Apple 授权地向公众公开了。在该明显盗窃之前,尚未提交本申请所基于的美国优先权申请。

[0002] 相关申请的交叉引用

[0003] 这是基于 2010 年 4 月 7 日提交的名为“Opportunistic Multitasking”、序列号为 61/321,616 的美国临时申请的国际申请,在此通过引用结合该美国临时申请的全部内容。

背景技术

[0004] 本公开一般涉及计算机科学领域。更具体地,本公开涉及一种用于改进个人电子设备中的用户体验和功率管理的技术。

[0005] 功率限制的手持设备(例如移动电话、个人娱乐设备和电子板(pad)计算机)相对于更大型的固定系统(例如台式机、工作站和笔记本电脑)是资源有限的。在这样的系统中,可用的计算能力确实不能支持大量并发进程/线程(process/thread)的执行而不显著地劣化用户体验和消耗设备有限的功率资源。按照这种认知,这些类型的设备的系统设计者传统上不支持用户应用级别的多任务处理。虽然这种方式具有将设备的有限功率资源消耗降到最低的好处,但它也限制了(通过用户应用)向用户提供交互式环境的能力。

发明内容

[0006] 提供用于个人电子设备的服务,通过这些服务支持后台处理或多任务处理的形式。所公开的服务允许用户应用利用后台处理而对前台进程的用户体验或个人电子设备的功率资源没有显著负面结果。所公开的服务的实施对执行中的用户应用可以是基本上透明的,并且在某些情况下,可以无需用户应用的明确协作而执行。为了实现所公开的多任务处理,许多操作约束可以被强制实施。这些约束的一个结果可能是,进程将不能在后台进行其如果在前台则可能能够做的工作。

[0007] 在一种服务中,前台用户应用在其从前台状态迁出的时候,被转换成非执行状态。在另一种服务中,后台进程在其被转换为非执行状态之前,被赋予最大的时间量来完成任务。在另一种服务中,音频应用(例如生成或记录音频信号的用户应用)被允许在后台执行,直到和除非其被用户暂停。在另一种服务中,针对用户应用被实例化(instantiate)的通信套接字(socket)即使在实例化它们的用户应用处于非执行状态时,也可以维持。能够利用这种服务的一种例示类型的应用是互联网协议电话(VOIP)用户应用。在一些其他的实施例中,用户应用在处于非执行状态时,被允许接收通知(例如位置事件)。每种被公开的服务依赖或使用一个或多个在实施时能干扰或阻止全时多任务处理操作的约束。也就是说,所公开的服务仅仅在不会明显干扰前台进程或过度使用个人电子设备的功率的时候,才允许多任务处理。

附图说明

- [0008] 图 1 示出根据一个实施例的说明性线程调度方案。
- [0009] 图 2 以流程图形式示出根据一个实施例的任务完成服务操作。
- [0010] 图 3 以流程图形式示出根据一个实施例的音频服务操作。
- [0011] 图 4 以流程图形式示出根据一个实施例的网络服务操作。
- [0012] 图 5 以框图形式示出根据一个实施例的个人电子设备。
- [0013] 图 6 以框图形式示出根据另一个实施例的个人电子设备。
- [0014] 图 7 以框图形式示出根据一个实施例的无线通信和位置处理单元。
- [0015] 图 8 以框图形式示出根据一个实施例的个人电子设备。

具体实施方式

[0016] 提供用于个人电子设备的服务,通过这些服务支持受限形式的后台处理或多任务处理。利用一种或多种所公开的服务结合所公开的操作约束,允许用户应用在后台执行,以便向用户提供对设备功率资源影响很小的鲁棒的交互式环境。在某些实施例中,后台处理可被限制为完成某个具体任务、规定的时间量(例如 5-30 分钟)、或特定的任务类型(例如音频操作)。在其他实施例中,用户应用可以在通知操作系统其可能会在一个或多个指定事件发生时复活(reanimate)之后,停止操作。一旦复活,该应用可以在后台执行附加的操作。这些公开的服务的实施对执行中的用户应用可以是基本上透明的,并且在某些情况下,可以无需用户应用的明确协作而执行。

[0017] 如此处所用,短语“个人电子设备”是便携式数字设备,诸如移动电话、个人数字助理、手持娱乐设备、电子板(pad)或平板(tablet)计算机系统或机顶盒(例如:Apple TV[®]或有线转换盒)。(APPLE TV 是苹果公司的注册商标。)如此处所用,术语“服务”指的是执行某种用于调用进程的的任务的应用程序、函数或程序代码模块,其没有用户接口,并且通过诸如应用编程接口(API)等调用接口以程序方式被访问。

[0018] 在接下来的描述中,出于解释的目的,许多特定细节被阐述以便提供对本发明的详尽理解。然而,对本领域技术人员而言显而易见的是,本发明可以无需这些特定细节而实现。在其他情况下,为了避免模糊本发明,结构和设备以框图形式表示。对不带下标的数字的引用被理解为引用对应于该被引用数字的所有带下标的实例。此外,本公开中使用的语言主要是为了可读性和指导的目的而被原理性选择的,并且可能未被选择为描绘或限制创造性主题,有必要借助权利要求来确定这样的创造性主题。说明书中对“一个实施例”或“实施例”的提及意味着,结合实施例描述的特定属性、结构或特征被包括在发明的至少一个实施例中,并且对“一个实施例”或“实施例”的多次提及不应当被理解为一定都指同一个实施例。

[0019] 将理解的是,在任何实际实施方式的开发中,必须使用大量的编程和元件决策来完成开发者的特定目的(例如顺应与系统和商业相关的约束),并且这些目的从一种实施方式到另一实施方式将是不同的。同样将理解,这种开发工作可能是复杂和耗时的,但是对具有本公开优点的个人计算设备开发领域的普通技术人员而言,不过是一个常规工作。

[0020] 为了实现后台处理或多任务处理,而不给前台进程的用户体验或个人电子设备的功率资源带来显著的负面结果,可能会强制实施许多操作约束。这些约束的一个结果可能是,进程将不能在后台进行其如果在前台则可能能够做的工作。

[0021] 如此处所用,短语“前台进程”表示当前可以访问系统资源(例如平台的中央处理单元和图形处理单元)并且向用户呈现用户界面(UI)或图形用户界面(GUI)的进程。(相应地,“前台”是前台进程执行时所处的状态。)相对地,“后台进程”是可被调度为访问系统资源但是不把UI/GUI当即呈现给用户的进程。(相应地,“后台”是后台进程执行时所处的状态。)如此处所用,术语“进程”表示用户应用。而用户应用是能够呈现UI/GUI的可执行的代码模块。

[0022] 操作约束:接下来的操作约束的说明性列表可以被强制实行,以限制后台进程可能会对前台进程产生的有害影响。

[0023] 1 处理器调度 (Processor Scheduling)。以后台进程不干扰前台进程的方式,向前台和后台进程分配不同的处理器调度“优先级”。本领域的普通技术人员会认识到,在像 **UNIX**[®]和 Mach 这样的操作环境中,不是调度进程,而是调度线程。(UNIX 是美国电话电报公司的注册商标)。进一步可以理解的是,线程是程序、应用或进程的一部分,这部分可以与该程序、应用或进程的其他部分独立地和并行地运行。

[0024] 参照图 1,在一个实施例中,后台线程可以从与分配给前台线程的那些优先级别部分重叠的优先级范围(band)中,被指定一个优先级。该重叠的优先级范围增加了后台进程/线程取得进展(即,执行)的可能性,即便此时有前台进程/线程正在运行。在该实施例中,为了提高线程的响应度,对使用它们的整个时间片(quantum)而未阻断输入/输出(I/O)操作的那些线程进行降级(即,降低它们的调度优先级),被发现是有益的。在一个实施例中,每次线程无阻断地使用其全部的配给时间片时,线程的优先级降低单个计数。(将认识到,进程使用全时间片表明该进程正在使用中央处理单元。)在另一个实施例中,当这种情况发生的时候,线程优先级可以被降低多于一个计数。优先级在前台范围和后台范围之间“重叠”的量可以是一(1)个或多个。

[0025] 2 盘调度 (Disk Scheduling):在一个实施例中,当访问系统存储单元(例如磁盘驱动器和固态硬盘单元)时,前台线程相对于后台线程被给予优先权。此外,后台进程(即,与处于后台状态的进程相关联的线程)在其访问系统存储单元时可以被限制速率。

[0026] 3 进入网络活动 (Incoming Network Activity)。在一个实施例中,进入网络通信量可以通过丢弃送往处于后台或非执行状态(例如挂起(suspended))的应用的所有包而得到缓和。进一步地,人为地小的缓冲区大小可被报告给末端网络站点,以便抑制进入的TCP(传输控制协议)通信量。本领域的普通技术人员将知道,TCP 具有现有的一套与缓冲区大小相关的通信流量控制机制,从而通信量不会以比接收方能处理的更快的速度通过网络被发送。如前面所述,在一个实施例中,人为地小的缓冲区大小可以被报告,使得发送方减慢或者停止进入的通信流量。

[0027] 4 外出网络活动 (Outgoing Network Activity)。在一个实施例中,网络访问通过使用两(2)个队列而被调解;一个用于与前台进程相关联的作业,一个用于与后台进程相关联的作业。赋予前台队列优先于后台队列被发现是有益处的。举例而言,前台队列可以被赋予100%优于后台队列的优先级。在这种设置下,后台队列中的作业仅有的被服务的时间就是当前台队列空了的时候。在另一个实施例中,前台队列里的作业与后台队列里的作业被服务的比率可以是任何希望的比率(例如,90/10,80/20,75/25或50/50)。在另一个实施例中,一旦作业已被放置到队列(例如,前台队列)之中,则如果该作业的相关进程随

后被置于一个不同的操作状态（例如，后台），它也不会移动。在另一个实施例中，作业的队列位置设更新，以反映其相关进程的操作状态。

[0028] 5 图形操作 (Graphics Operations)。在一个实施例中，比如举例来说，iPhone® 操作系统，至少有两个机制，应用可以通过这些机制对显示屏写或画（IPHONE 是苹果公司的注册商标）。第一个机制是通过使用专门的硬件，比如图形处理单元（GPU）。第二个机制是通过使用一个或多个平台中央处理单元（CPU）。依照一个实施例，在 iPhone 的操作环境中，后台进程不被允许访问专门的图形硬件；例如，任何企图访问 GPU 的后台进程都可被终止。再举例而言，试图通过 CPU 生成可视化图形的后台进程被忽视——也就是说，执行意图利用 CPU 写图形缓冲区 / 内存的命令的任何尝试都会被忽视。在一些实施例中，与置于后台状态的进程 / 应用相关联的图形内存可被标记为“可清除的 (purgeable)”。因此，如果前台进程需要更多的内存，它可以使用（通过标准内存分配技术）原本授予当前处于后台状态的进程的显存。

[0029] 6 通知 (Notifications)。当进程从后台状态转换到前台状态时，允许其为每种类型的通知接收一 (1) 个通知。更特别地，在一个实施例中，多个通知事件可以被合并。例如，如果用户将个人电子设备（例如 iPhone）从纵向显示旋转到横向显示、到纵向显示、然后回到横向显示，则产生的三 (3) 个方向通知事件可以合并为一个通知——最后的状态，横向。这可以改善应用的响应，因为不需要响应多个事件。示例性的通知类型包括方向改变通知（例如，基于加速度计或陀螺仪传感器输入）、通讯录数据库通知（当通讯录条目被修改时发生）和摄像头滚转通知（当共享资源图像库中的图像已被改变时发生）。

[0030] 7 锁 (Locks)。在一个实施例中，任何即将进入挂起状态的在共享资源上具有锁的进程都可被终止（而不是挂起）。示例性的共享资源包括专门的图形硬件和文件描述符。在另一个实施例中，当用户应用进入后台状态时，共享资源从用户应用中被“拿走”。例如，如果应用（进程）在摄像头硬件上有锁，则当用户应用被置于后台状态时，该锁被移除。如此处所用，“挂起状态”是进程不被允许调度线程以供执行，但是其状态保留在主存中的情况。

[0031] 8 硬件约束 (Hardware Restrictions)。在一个实施例中，后台进程被阻止访问某些系统硬件资源。后台进程不可用的示例性的硬件包括但不限于：摄像头、GPU、加速度计、陀螺仪、接近传感器和麦克风。

[0032] 9 内存管理 (Memory Management)。为了保持足够的内存供给执行的进程，当个人电子设备的核心内存接近一个指定的“临界”点时，进程可从内存中移除。在一个实施例中，这样移除的进程可以休眠。在另一个实施例中，这样移除的进程可以被终止。如此处所用，短语“核心内存”或“主存”表示从中执行用户应用的个人电子设备的主要随机存取存储器；“休眠进程”是其操作状态已写入非易失性存储器的进程，从而 (1) 其不能调度线程以供执行，以及 (2) 其主存的“足迹 (footprint)”基本上为零；而“终止进程”是其操作状态已被删除并且因此直到重启才能被执行的进程。在一个实施例中，在休眠动作期间，可根据需要重新创建（可清除的）或重新载入（干净的）的内存被清除；只有已被修改或不可重新构建的内存才被写入非易失性存储器中。在另一个实施例中，非易失性存储器的指定区域在引导时被留出，用来作为“休眠”存储区。当休眠进程复活时，在非易失性存储器中的其数据被送回到主存中，从而在非易失性存储器中释放空间。在一个实施例中，未被访问超过指

定次数的应用可能被“休眠”。

[0033] 可以采用任何希望的方式选择哪些应用要从主存中被移除。例如,最近最少使用的应用可被首先移除。可替换地,最大的当前非执行应用可被移除。在一个实施例中,被选择移除的大于指定大小(例如,8MB、16MB、或 32MB)的应用可被终止,而小于该指定大小的应用可被休眠。在另一实施例中,后台音频用户应用可以仅在挂起的用户应用已被终止之后、但是在前台用户应用之前被终止。

[0034] 类似地,“临界”点可以是设计者希望的任何值。示例性的临界点包括 50%、75% 和 90%。若采用 75%作为示例的临界点,一旦主存达到其容量的 75%(例如,当 256 兆字节(MB)的主存达到满 192MB 的时候),将从主存中选择应用以便移除。

[0035] 操作状态:为了帮助理解以上约束是如何应用到以下描述的各种服务的,表 1 标识和描述了在个人电子设备上执行的用户进程的五(5)种操作状态。

[0036] 表 1. 操作状态

[0037]

模式 或 状态	描述	约束
前台	处于这个状态的进程可以访问系统资源并且向用户当前展示用户界面。	无
后台	处于这个状态的进程可以访问系统资源但是不能向用户当前展示用户界面。	实时进程受制于实时约束(即,当置于后台时,它们被降级为非实时优先级——除了音频和 VOIP/网络通信应用); 所有其他进程受制于所有其他约束。
挂起	处于这个状态的进程在主存中具有操作状态,并且若不是因为已被挂起,原本能够被调度执行。	无资格执行
休眠	处于这个状态的进程,其操作状态已被写入非易失性存储器。只有识别、定位和复活该进程所必需的信息才保留在主存中。	无资格执行
终止	其操作状态在主存中已经不再可访问的进程	无资格执行

[0038] 如此处所用,术语“非执行状态”表示挂起、休眠或终止状态之一。

[0039] 示例性的服务:在下面,所描述的服务进行操作所处的“环境”是 iPhone 操作系统。这仅是示例,所公开的服务可以等价地应用于任何个人电子设备。

[0040] 1. 快速任务切换服务 (Fast Task Switching Service)。在过去,在 iPhone 操作系统中,当应用离开前台状态时,其简单地被终止。快速任务切换服务允许应用接收其即将离开前台状态的通知,并且在这时被赋予进行某种处理的机会(例如:几秒)。在这个时间之后,该应用被挂起(即,被置于挂起状态)。在一个实施例中,依照本公开的操作环境可以对所有执行中的进程应用快速服务切换(除非利用如下所述的另一种服务)。在另一个实施例中,每个应用都必须以程序方式规定其可以被挂起。

[0041] 当挂起时,进程/应用不能执行指令。在像 Unix 或 Mach 这样的线程操作环境中,这意味着进程不能调度线程以供执行。然而在某些实施例中,代表进程而启动的内核工作(kernel efforts)即便在进程本身已被挂起(或休眠或终止)之后也可以继续执行。示例的“内核工作”包括进行中的 I/O 操作和虚拟内存和网络缓冲操作。这些操作的完成可能会或不会引发挂起进程的复活。

[0042] 需要注意的是,在此讨论的这种类型的进程挂起不同于系统级别的操作,诸如现有技术中的“睡眠”或“休眠”动作——两种都是“系统范围”的操作。根据现有技术,没有进程可以在睡眠或休眠模式下执行。相反,这里描述的挂起模式应用到单独的进程;第一进程的挂起不会干扰第二进程的执行。

[0043] 2. 任务完成服务 (Task Completion Service)。任务完成服务将有限的能力提供给进程,来完成处于后台状态的以前台状态开始的任务。在一个实施例中,后台进程(即,以后台状态执行的进程)被抑制以避免干扰前台进程。良好适应于使用这种服务的示例性任务是网络上传和下载操作、盘存储和检索操作、以及图像处理操作(例如,那些不需要像 GPU 这样的专门图形硬件的操作)。在一个实施例中,后台进程被允许继续处理(即,如上所述地接收与其调度优先级一致的 CPU 时间)直到特定事件发生。示例性的事件包括系统定时器(给予该进程指定的时间量来完成处理,例如 5 到 30 分钟的“时钟时间”、或 5 毫秒的 CPU 时间)、设备外部电源的失去、或者主要或次要显示器的激活。

[0044] 参照图 2,示例的任务完成操作 200 在发现非前台应用(第二应用)将进入前台(块 205)时开始。这可以由例如以下引起:用户请求开始第二用户应用、转换到系统应用、或对 VOIP 用户应用的进入呼叫。如果当前前台用户应用(第一应用)没有将其本身指定为能够在后台执行的应用(块 210 的“否”箭头),则第一应用被置于非执行状态(块 215)且第二应用进入前台(块 220)。举例而言,第一应用可以被挂起(即,置于挂起状态)。如果第一应用已经将其自身指定为能够在后台执行的应用,例如通过 p 列表(p-list)条目(块 210 的“是”箭头),则第一应用被置于后台状态且被赋予指定时间量来完成其当前操作(块 225)。第二应用然后可以进入前台并被允许执行(块 230)。一旦处于后台,第一应用可以适时地被调度执行(参见以上关于处理器调度的讨论)(块 235)。在指定事件发生前且其未完成任务时,第一应用被允许在后台执行。一旦该事件发生或进程完成其任务(块 240 和 245 的“是”箭头),第一应用可以被置于非执行状态;例如,挂起或终止状态。如果这些事件无一发生(块 240 和 245 的“否”箭头),则第一应用继续执行(块 235)。

[0045] 在一个实施例中,不活动的前台进程访问全部系统资源的能力可以类似地被限制。例如,一个或多个不活动的前台进程(即,当前不在接收用户输入的前台进程)对存储盘(约束 2)、进入和外出网络缓冲区(约束 3 和 4)的访问和/或与指定硬件(约束 8)交互的能力可以以这里针对后台进程所描述的相同方式而被限制。以类似的方式,考虑到功

率损耗,除了活动的前台进程以外的所有前台进程都可以被挂起或“推”到后台,之后它们的行为按照这里描述的后台处理方式被控制。

[0046] 在一个实施例中,进程必须以程序方式声明其想要利用任务完成服务(例如,在根据块 210 的动作期间)。这可以这样实现:例如,静态声明用户应用可以通过在安装时或第一次运行时由用户指定的 API 调用、p 列表条目、或可变的设置来利用任务完成服务。在另一个实施例中,基于应用在运行时以代码形式声明其要开始操作(随后接着声明其已经结束了该操作)而准予任务完成。如果这样的任务在该应用离开前台状态时仍然未完成,应用可以被准予更多的时间来完成它们。在指定事件发生后,或任务完成不久,应用可被挂起。在另一个实施例中,执行指定数量的合格任务之一的任何进程或用户应用可以利用任务完成服务(例如,前述操作中的一个)。例如,执行文件上传或下载操作的任何进程可以根据任务完成服务约束被放置到后台。

[0047] 3. 音频服务(Audio Service)。音频服务允许进程或应用在处于后台状态时播放或记录音频。在一个实施例中,音频进程利用实时线程并被相应地调度(参见图 1)。虽然一般而言,实时进程在置于后台时被降级到非实时优先级,但是对于使用音频服务的应用而言并不是这样。因为描述的实施例为音频应用使用实时线程,为了最小化与前台进程之间的干扰,可以一次只允许一个实时音频进程处于后台状态。(将认识到,这个数字仅仅是示例。如果个人电子设备的处理能力足够,则同一时间可以有多个音频进程在后台活动。)

[0048] 参照图 3,当执行音频操作(例如,音乐重放或语音记录操作)的前台用户应用接收到关于另一个当前不在前台的应用(第二应用)将进入前台的通知(块 305)时,示例的音频服务操作 300 开始。该用户的音频应用随后被置于后台(块 310),并且第二应用进入前台(块 315)。在一个实施例中,实时后台线程可以被限制为从指定的一组库中执行。例如,实时后台线程可以被阻止使用第三方库(例如,应用开发者支持的库)。在将来的某时候,用户可以向其后台音频应用发出“暂停”命令(块 320)。作为响应,用户的后台音频应用从后台状态移到挂起状态(块 325)。音频应用保持在挂起状态,直到用户发出“恢复(resume)”命令(块 330 的“是”箭头),此后该音频应用移回到后台状态,在后台状态其可以继续执行其音频操作(块 335)。在另一个实施例中,音频应用可被置于休眠模式而不是挂起模式。

[0049] 在某些实施例中,进程可以静态声明其能够通过安装在安装时或第一次运行时由用户指定的 API 调用、p 列表条目、或可变的设置来利用音频服务。在其他实施例中,任何播放或记录音频的进程可以将音频服务应用到其上。当执行这些应用中的状态码时,根据快速任务切换或任务完成服务,关联的进程在音频操作期间不能被挂起。

[0050] 4. 通知服务(Notification Service)。通知服务允许任何状态(前台、后台、挂起或休眠)下的进程接收和响应消息。这些消息可以源自外部源(“推送”通知)或接收进程/应用本身(“本地”通知)。

[0051] 通常而言,一旦接收到消息,操作系统模块(例如 iPhone 操作系统中的“跳板(Springboard)”应用)识别目标应用(即,通知消息指向的应用),在后台启动该应用,并且将消息传送给这个目前复活的应用。在另一个实施例中,一旦接收到消息,操作系统(例如跳板)产生对话框,通过对话框,用户可以选择切换到目标应用(即,通知消息指向的应用)。如果这样选择,则目标应用进入前台。(将认识到,如果目标应用已经在后台,则它不

需要被复活。)然后,该应用被赋予响应消息的机会,包括通过例如操作系统请求向用户呈现对话。然后根据相关服务(例如,快速任务切换、任务完成、音频、VOIP、或位置和地理围栏(Geo-Fencing))提供对接收进程的控制。在另一个实施例中,前往后台进程的通知可以被排队并且在稍后(例如:每5分钟)被传送,而不是在接收到消息后立即传送。

[0052] 5. 网络服务(Network Service)。网络服务允许挂起的网络应用维持(即,保持活跃)其通信套接字。如此处所用,网络应用是使用网络通信套接字的用户应用。示例的网络应用是用户 VOIP 应用。在现有技术中,当用户的网络应用/进程挂起时,它失去了维持其通信套接字的能力,结果就不能继续接收通信数据(例如,进入的 VOIP 电话呼叫)。

[0053] 参照图 4,示例的网络服务操作通过 VOIP 服务操作 400 来阐释,VOIP 服务 400 从用户的 VOIP 应用转换到挂起状态(块 405)开始。在一个实施例中,当 VOIP 进程在被转换到挂起状态时,它向操作系统(例如通过网络守护程序(daemon))指定需要什么时间间隔来发送“心跳”消息(410)。该守护程序负责当在与进程相关的套接字处接收到数据或指定时间到期时(后者是为了维持心跳)复活进程。关于这个方面的更多信息,参见名为 Managing Notification Service Connections and Displaying Icon Badges 的美国专利公开 20090305732。该申请全部被结合到此。

[0054] 在稍后的某一时刻,该网络守护程序接收到消息(块 415)。然后目标应用被识到(块 420)、复活到后台状态(块 425)、并被传送消息(块 430)。后台 VOIP 应用然后可以响应该消息(例如,进入的 VOIP 呼叫或“需要心跳”消息)(块 435)。如果需要的任务是发出心跳,该 VOIP 应用可以返回到挂起状态。如果需要的任务是响应进入的 VOIP 呼叫,该 VOIP 应用可以请求操作系统发出 UI,从而用户可以选择接收或忽略该呼叫。如果呼叫被忽略,该 VOIP 应用可返回到挂起状态。如果呼叫更被应答,该 VOIP 应用可以进入前台(例如,借助于用户在操作系统呈现的 UI 中选择适当的元素)。虽然将认识到操作系统守护程序在本领域中是公知的,但是在此描述的这种类型的在个人电子设备中为挂起的进程维持网络通信(例如 VOIP 和无线保真或 WiFi)套接字是新颖的。

[0055] 6. 位置和地理围栏服务(Location and Geo-Fencing Service)。位置和地理围栏服务允许后台或挂起进程在指定位置事件发生时接收通知(例如,来自操作系统守护程序的消息)。示例的位置事件包括但不限于,到达指定地点、离开指定地点、进入指定区域、离开指定区域、或移动“明显”距离(其中什么算“明显”可以由用户或用户应用指定)。在一个实施例中,进程/应用必须以程序方式声明其想要利用位置和地理围栏操作服务(例如通过 p 列表参数)。在另一个实施例中,地理围栏操作不需要利用 p 列表条目——所有的地理围栏应用都可以利用地理围栏服务。

[0056] 在位置操作中,只要位置跟踪操作在进行,应用就可以保持在后台——接收依照系统调度机制(参见上面讨论)的处理时间。在一个实施例中,当跟踪操作终止或中断时,该进程可被挂起(即,置于挂起状态;进程也可以被终止或休眠)。一旦进程响应了消息,它可以停顿(quiesce)(返回到挂起状态)或留在后台状态等待下一个消息。在另一个实施例中,在后台状态使用位置服务的用户应用的数量与其他系统资源的利用有关;例如,如果系统内存运行慢,则当前执行位置跟踪的用户应用可以被终止。在另一个实施例中,集中式守护程序集合所有的位置请求(特别是关于所请求的期望精度),并且针对所请求的最激进(aggressive)的模式启动适当的硬件。例如,如果一个用户应用请求达 3 公里以内(这

种情况下,可以采用基于移动电话小区的定位)的用户位置,第二个用户应用请求达 500 米以内(这种情况下,可以采用 WiFi 扫描)的用户位置,以及第三个应用请求达 100 米以内(GPS 对其是必需的)的用户位置,则可以使用 GPS 定位,并且在后台的所有跟踪定位应用都将获得该精细定位的益处。

[0057] 在地理围栏操作中,只要穿过指定的区域边界,进程就接收到来自操作系统的消息(例如通过网络守护程序)。如果进程正在执行(例如,在后台,并且在其分配的时间期间),它会被赋予处理该消息的机会。如果进程不是正在进行处理(例如,其在后台并且没有在进行处理,或者被挂起或休眠),则该进程/应用将复活,此后它会被允许处理该事件(在后台)。此外,如果进程已经被终止,它可以以与先前描述的通知服务类似的方式复活进入后台状态。参见名为“Location-Based Application Program Management”、序列号为 12/123,456 的相关公开。在位置处理使用单独处理器而不是使用为用户应用执行运行操作系统的处理器(此后称为“应用处理器”)的实施例,这里描述的类型的位置和地理围栏操作即使在应用处理器处于睡眠时也可以发生。与位置服务一样,后台状态中可以利用地理围栏服务的用户应用的数量与其他系统资源的利用有关;例如,如果系统内存运行慢,则当前执行位置跟踪的用户应用可以被终止。

[0058] 个人电子设备结构:参照图 5,在一个实施例中,个人电子设备 500 可利用一个或多个上述约束实施一个或多个上面的服务,以便在这样做有益的时候允许多任务处理操作(例如,机会性的多任务处理)。如所示的,用户应用 505A 到 505N 与用于标准操作系统调用的应用启动器(application launcher)510(例如 iPhone 操作系统中的“跳板”)和多任务处理服务库 515 交互,以便访问和利用上面描述的服务。

[0059] 参照图 6,根据另一个实施例的系统 600 也可以包括无线电装置 605 和位置处理装置 610。如图 7 中所示,无线电装置 605 可以包括接收器电路 700,发射器电路 705 和存储器 710。接收器和发射器电路 700 和 705 可以用于连接到例如移动电话网络、卫星通信网络或数字计算机网络(例如,通过局域网 WiFi 连接的因特网)。存储器 710 可以用于存储接收的数据、要发射的数据和本领域已知的操作参数信息。接口 715 可以被设置以帮助无线电装置 605 和位置装置 610 之间的通信。位置装置 610 可以包括处理器 720、存储器 725 和操作系统接口 730。通常来说,位置装置 610 接收对位置更新和地理围栏限制的请求(例如,通过接口 730 从一个或多个用户应用 505A 到 505N 接收),处理从无线电装置 605 接收的定位信息(通过接口 715 接收),以及通过接口 730 向进行请求的应用返回位置信息和更新。处理器 720 可以是一个或多个现成的处理器、一个或多个定制设计的处理器、或现成的和定制设计的处理器的结合。存储器 725 可以是易失性和非易失性的固态、磁性或光学存储器的任意形式或结合。

[0060] 在一个实施例中,一个或多个所描述的服务可以基于电源的可用性被提供。例如,如果个人电子设备插入到电源(例如交流电源)上,则多任务处理能力可以根据标准现代操作系统方式(即,完全的或无约束的多任务处理)被提供。在另一方面,如果用户电子设备没有插入到电源上,则多任务处理能力可以根据本公开(例如,部分的或机会性的多任务处理)而被提供。

[0061] 参照图 8,个人电子设备 800 可以嵌入到包含处理单元 805 和支持设备 810 的物理设备中。处理单元 805 可以包括处理器 815、存储器 820 和输入-输出电路 825。设备 800

也可以包括所示的无线电装置 605 和位置装置 610。输入 - 输出电路 825 将处理器 815 和存储器 820 耦接到一个或多个显示单元 830 (例如, 触摸屏显示器)、一个或多个输入 - 输出设备 835 (例如, 触摸屏、键盘、鼠标、操纵杆)、一个或多个程序存储设备 840 (例如, 磁性、光学或固态存储器)、以及其他硬件 845 (例如, 网络接口)。注意, 在图 6、7 和 8 呈现的结构中, 即使当 / 如果作出位置或地理围栏请求的应用 (例如应用 505A) 已经被挂起、休眠或终止时, 位置和地理围栏处理也可以通过无线电装置 605 和位置装置 610 继续。更进一步地, 即使在应用处理器 815 休眠或睡眠时, 位置和地理围栏操作也可以继续。也就是说, 位置和地理围栏操作不管应用处理器 815 处于什么状态, 都可以通过处理器 720 继续。

[0062] 材料、组件、电路元件的各种变化以及所示的操作方法的细节都是有可能的, 不会背离所附权利要求的范围。依据本公开的多任务处理能力的提供可以通过用于执行被组织为一个或多个程序模块的指令的可编程控制设备来完成。可编程控制设备 (例如处理器 720 和 215) 可以包括任何可编程的控制器设备, 包括例如英特尔公司的 Intel **Atom**[®]、**Core**[®]、**Pentium**[®] 和 **Celeron**[®] 处理器家族和 ARM 的 Cortex 和 ARM 处理器家族中的一个或多个成员、或定制设计的状态机。(INTEL、INTELATOM、CORE、PENTIUM 和 CELERON 是英特尔公司的注册商标。CORTEX 是 ARM 有限公司的注册商标。ARM 是 ARM 有限公司的注册商标。) 定制设计的状态机可以嵌入到诸如集成电路之类的硬件设备中, 所述集成电路包括但是不仅限于专用集成电路 (“ASIC”) 或现场可编程门阵列 (“FPGA”)。

[0063] 适合用于有形地嵌入程序指令的存储设备 (例如, 存储器 710、725、220 和 820) 包括但是不仅限于: 磁盘 (固定的、软盘和可移动的) 和磁带; 像 CD-ROM 和数字视频盘 (“DVD”) 这样的光学介质; 和像电可编程只读存储器 (“EPROM”)、电可擦除可编程只读存储器 (“EEPROM”)、可编程门阵列和闪存设备这样的半导体存储器设备。

[0064] 最后, 可以理解的是, 上述描述意在示例性, 而不是约束性。例如, 上述实施例可以结合彼此使用。对本领域技术人员而言, 在阅读了上面的描述后, 许多其他的实施例都将是明显的。因此, 本发明的范围应当参照所附的权利要求以及与这些权利要求所请求的权利相等同的全部范围而被确定。在所附的权利要求中, 术语 “包括 (including)” 和 “其中 (in which)” 分别作为术语 “包含 (comprising)” 和 “其中 (wherein)” 的简单英语等同词来使用。

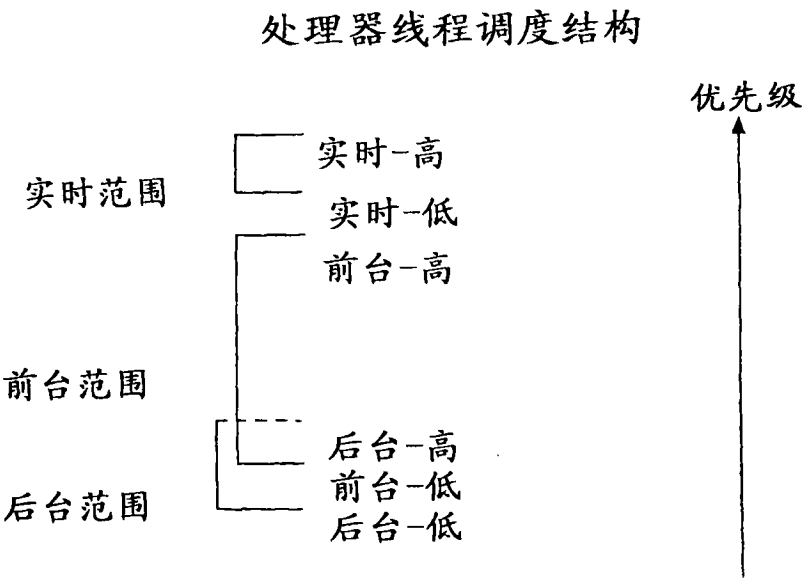


图 1

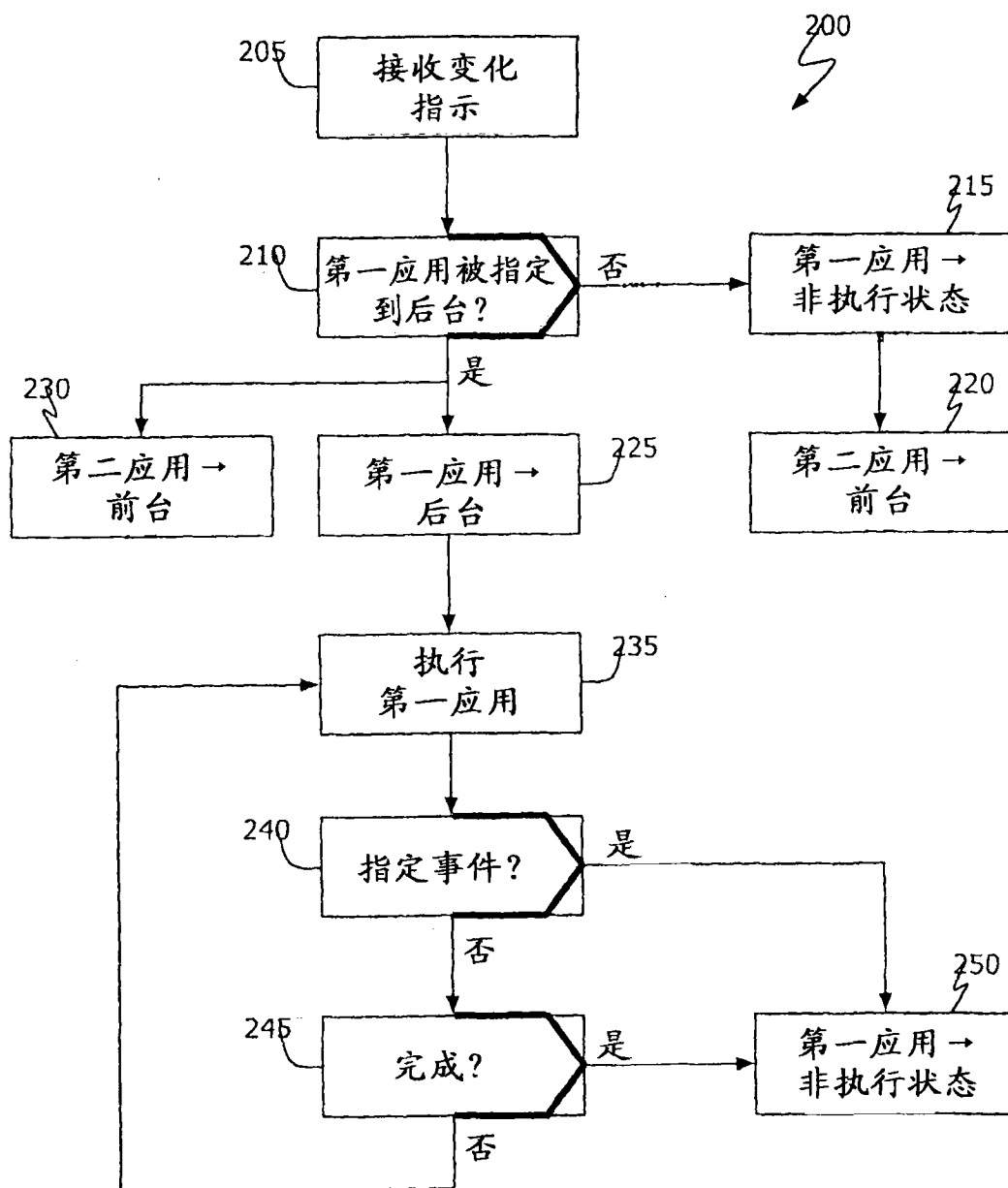


图 2

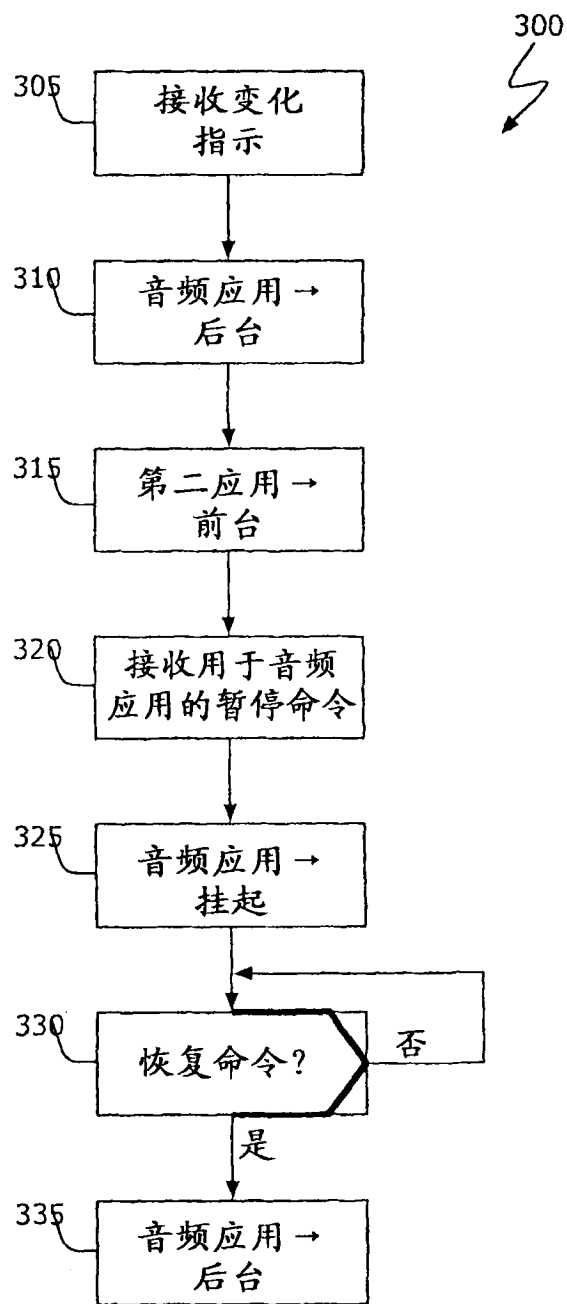


图 3

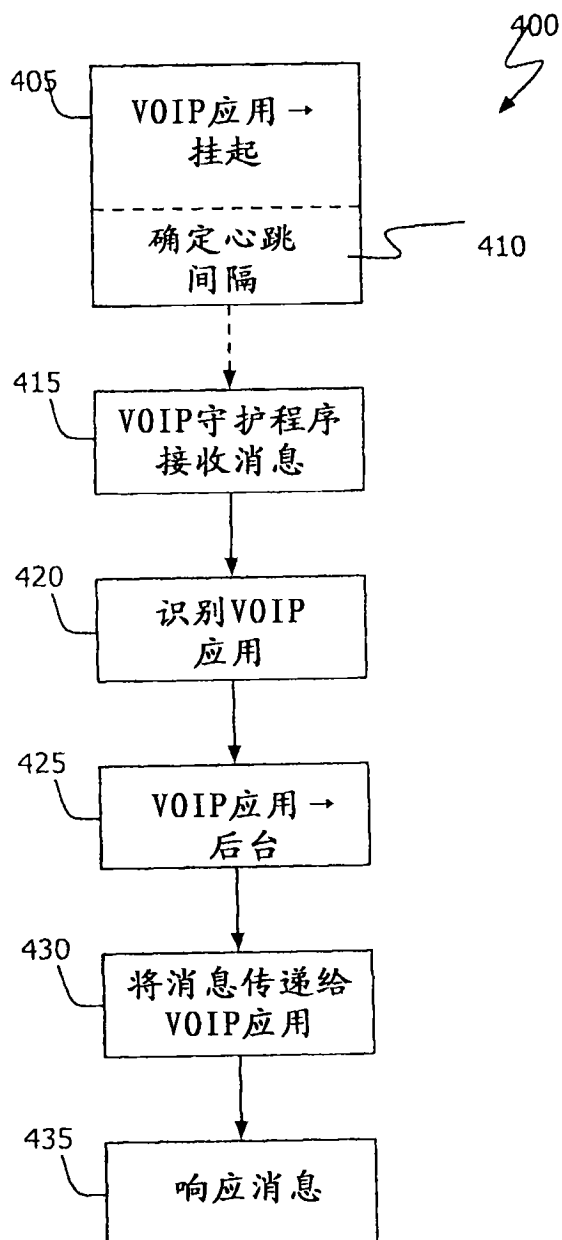


图 4

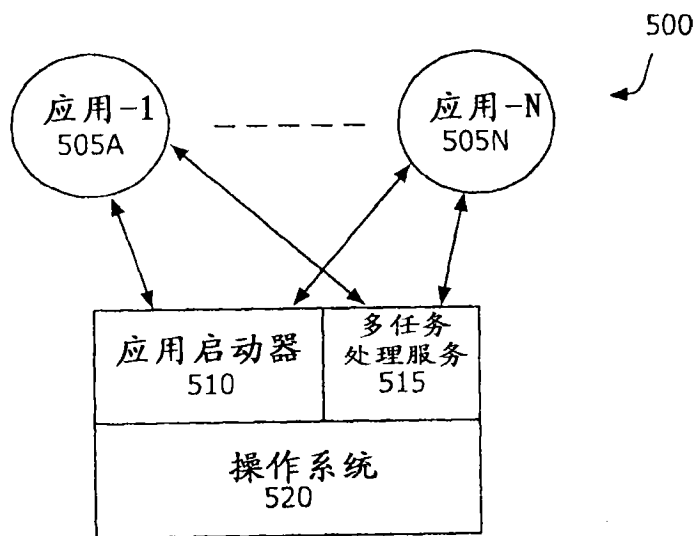


图 5

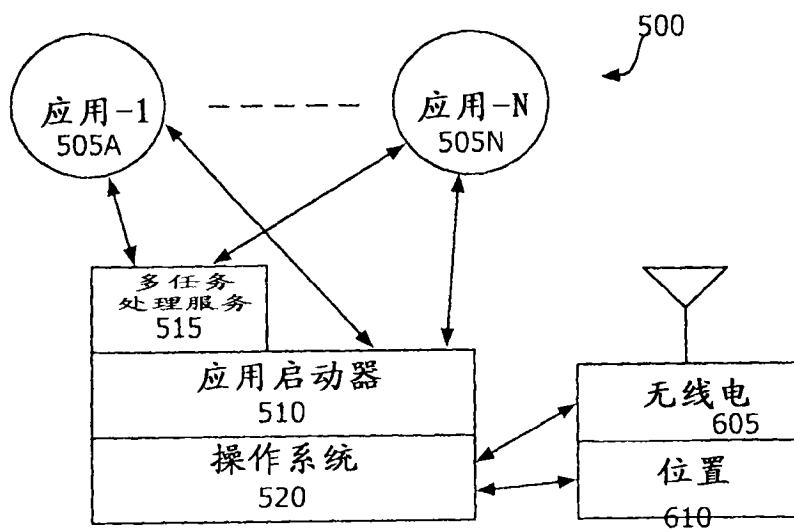


图 6

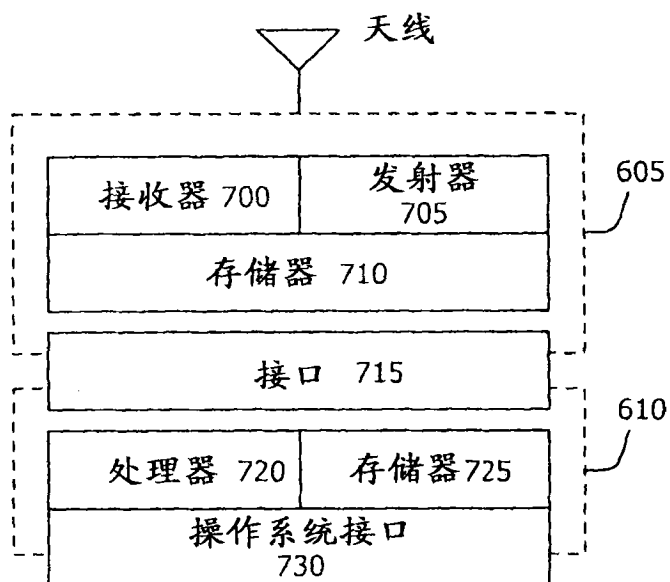


图 7

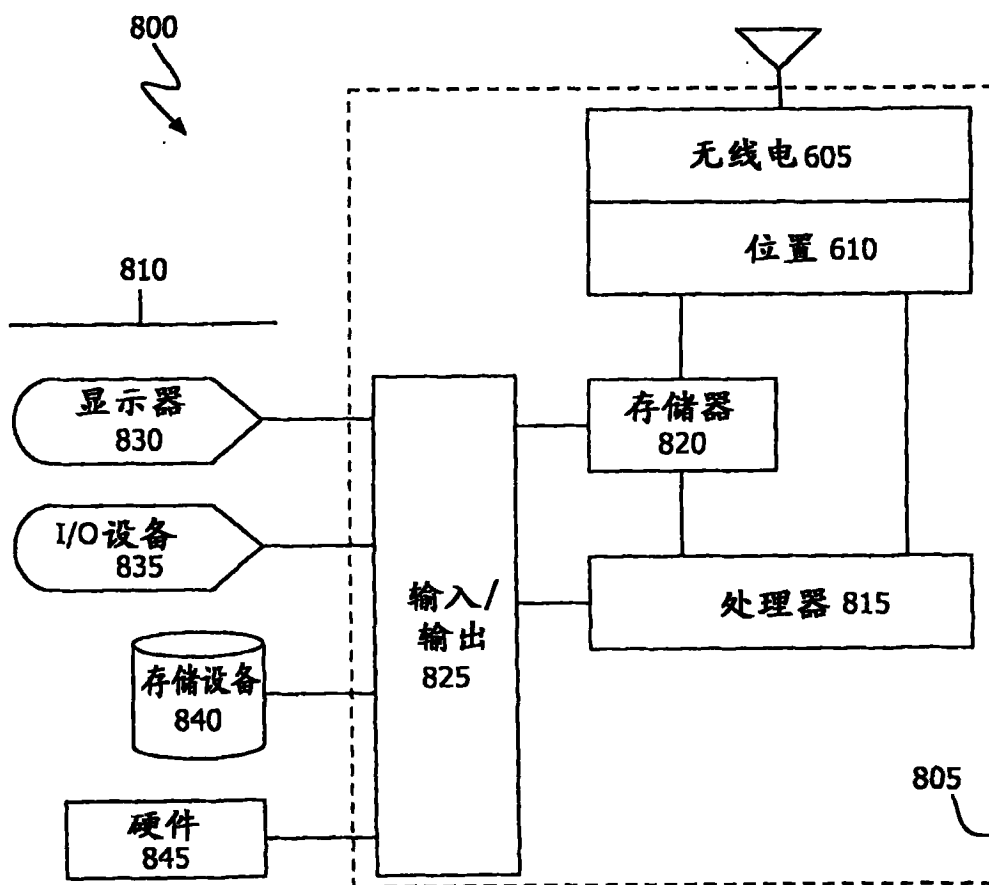


图 8