



(12)发明专利

(10)授权公告号 CN 106547628 B

(45)授权公告日 2020.05.01

(21)申请号 201611068613.4

(22)申请日 2016.11.29

(65)同一申请的已公布的文献号
申请公布号 CN 106547628 A

(43)申请公布日 2017.03.29

(73)专利权人 北京元心科技有限公司
地址 100176 北京市北京经济技术开发区
科创十四街99号33幢D栋2222号

(72)发明人 朱少峰

(74)专利代理机构 北京市立方律师事务所
11330

代理人 张筱宁 宋海斌

(51)Int.Cl.
G06F 9/50(2006.01)

(56)对比文件

US 2012324463 A1,2012.12.20,
CN 102129475 A,2011.07.20,
CN 103140832 A,2013.06.05,
CN 101894047 A,2010.11.24,
CN 105159770 A,2015.12.16,
CN 1264078 A,2000.08.23,
CN 103229156 A,2013.07.31,
CN 101154167 A,2008.04.02,
CN 101499021 A,2009.08.05,
吴义鹏.基于容器的虚拟机调度算法优化及实现.《中国优秀硕士学位论文全文数据库 信息科技辑》.2011,(第 09 期),第I138-1263页.

审查员 王佳

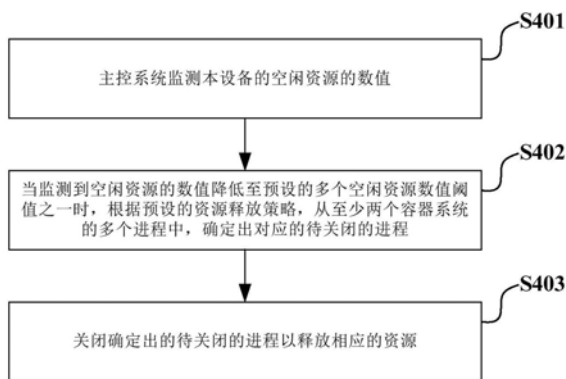
权利要求书2页 说明书16页 附图3页

(54)发明名称

多系统的资源释放方法及装置

(57)摘要

本发明实施例提供了多系统的资源释放方法及装置,所述方法包括:主控系统监测本设备的空闲资源的数值;当监测到空闲资源的数值降低至预设的多个空闲资源数值阈值之一时,根据预设的资源释放策略,从至少两个容器系统的多个进程中,确定出当前空闲资源数据阈值对应的待关闭的进程;关闭确定出的待关闭的进程以释放相应的资源。本发明实施例中,根据统一的资源释放策略统一评价多个容器系统的所有进程的重要度,根据统一的评价结果对多个容器系统的进程进行关闭以释放所占系统资源,实现对多个容器系统中的进程所占系统资源的管控;优化了多系统下的系统资源的释放效果,可以给用户带来便利,提升用户的体验。



1. 一种多系统的资源释放方法,其特征在于,包括:

主控系统监测本设备的空闲资源的数值;

当监测到所述空闲资源的数值降低至预设的多个空闲资源数据阈值之一时,根据预设的资源释放策略中多个空闲资源数据阈值与进程的综合重要度值的对应关系,确定出当前空闲资源数据阈值所对应的进程的综合重要度值,作为当前综合重要度值阈值;

当所述资源释放策略中进程的综合重要度值与进程的重要度的关系为综合重要度值越大则进程的重要度越小时,从至少两个容器系统的多个进程中,确定出综合重要度值大于所述当前综合重要度值阈值的进程,作为所述当前空闲资源数据阈值对应的待关闭的进程;

关闭确定出的待关闭的进程以释放相应的资源;

所述进程的综合重要度值是根据下述方法确定的:

每个容器系统根据所述资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值后,向所述主控系统发送;

所述主控系统根据所述资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出每个容器系统的系统重要度值,作为该容器系统中进程的系统重要度值;

所述主控系统对于每个容器系统中的每个进程,根据该进程的基础重要度值和系统重要度值生成该进程的综合重要度值。

2. 根据权利要求1所述的方法,其特征在于,所述进程的综合重要度值是根据下述方法确定的:

每个容器系统根据所述资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值;

每个容器系统根据所述资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出该容器系统的系统重要度值,作为该容器系统中进程的系统重要度值;

每个容器系统对于其中的每个进程,根据该进程的基础重要度值和系统重要度值生成该进程的综合重要度值后,向所述主控系统发送。

3. 根据权利要求1或2所述的方法,其特征在于,所述根据该进程的基础重要度值和系统重要度值,生成该进程的综合重要度值,包括下述至少一项:

将该进程的基础重要度值与该进程的系统重要度值进行累加,得到该进程的综合重要度值;

根据该进程的系统重要度值对该进程的基础重要度值进行加权,得到该进程的综合重要度值。

4. 根据权利要求1或2所述的方法,其特征在于,所述关闭确定出的待关闭的进程,包括:

当确定出多个待关闭的进程时,根据各待关闭的进程的综合重要度值从大到小的顺序,依次关闭待关闭的进程。

5. 根据权利要求1或2所述的方法,其特征在于,还包括:

依照基础重要度值从大到小的顺序,与基础重要度值对应的进程类型分别为:空闲进程、后台隐藏应用进程、后台服务、桌面应用进程、服务、备份应用进程、高负载应用进程、可感知应用进程、可见应用进程、前台应用进程、驻留程序进程、系统进程。

6. 一种多系统的资源释放装置,其特征在于,包括:主控系统和至少两个容器系统;
所述主控系统包括:

空闲资源监测模块,用于监测本设备的空闲资源的数值;当监测到所述空闲资源的数值降低至预设的多个空闲资源数据阈值之一时,发送资源释放通知;

待关闭进程确定模块,用于接收到资源释放通知后,根据预设的资源释放策略,从至少两个容器系统的多个进程中,确定出当前空闲资源数据阈值对应的待关闭的进程,具体用于根据所述资源释放策略中多个空闲资源数据阈值与进程的综合重要度值的对应关系,确定出当前空闲资源数据阈值所对应的进程的综合重要度值,作为当前综合重要度值阈值;当所述资源释放策略中进程的综合重要度值与进程的重要度的关系为综合重要度值越大则进程的重要度越小时,从至少两个容器系统的多个进程中,确定出综合重要度值大于所述当前综合重要度值阈值的进程,作为所述当前空闲资源数据阈值对应的待关闭的进程;

资源释放模块,用于关闭确定出的待关闭的进程以释放相应的资源;

每个容器系统包括:

基础重要度值确定模块,用于根据所述资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值后,向所述主控系统发送;以及

所述主控系统还包括:

系统重要度值确定模块,用于根据所述资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出每个容器系统的系统重要度值,作为该容器系统中进程的系统重要度值;

综合重要度值确定模块,用于对于每个容器系统中的每个进程,根据该进程的基础重要度值和系统重要度值生成该进程的综合重要度值。

多系统的资源释放方法及装置

技术领域

[0001] 本发明涉及终端技术领域,具体而言,本发明涉及一种多系统的资源释放方法及装置。

背景技术

[0002] 随着社会的进步和科技的发展,智能手机、平板电脑和电子阅读器等终端设备已广泛普及。相当一部分数量的终端设备中安装了两个操作系统。

[0003] 出于系统快速切换的需要,两个操作系统通常是同时运行在终端设备中的。在这种情况下,相比于运行单个操作系统,终端设备系统需要提供更多的系统资源,例如处理器资源和内存资源等,以保证两个操作系统的正常运行。

[0004] 在终端设备上,为了保证用户使用应用流畅的体验,当一个应用从前台切换到后台的时候往往不会释放该应用所占用的系统资源,而是继续占用原先的系统资源。这样当新的应用被打开并且申请系统资源的时候,终端设备会从空闲的资源中分配相应的系统资源供新的应用使用。经过一段时间的运行,终端设备往往容易出现空闲资源不足的情况,这就需要关闭那些当前还继续占用系统资源的应用的进程以释放相应的系统资源。现有的双系统还未配置行之有效的资源释放方法。

[0005] 现有的单系统的资源释放方法包括:选择相对不重要的应用的进程进行关闭,以释放相应的资源。然而,若把单系统的资源释放方法直接运用到双系统中的终端设备中,很可能该资源释放方法仅能对其中一个操作系统起到作用,而对其它操作系统不起作用。相当于仅有一个操作系统会产生资源释放的效果,导致系统资源的实际释放效果相当有限,在双系统占用系统资源明显多于单系统的情况下,容易导致配置双系统的终端设备经常出现系统资源不足的局面,给用户带来不便,降低了用户的体验。

发明内容

[0006] 本发明针对现有方式的缺点,提出一种多系统的资源释放方法及装置,用以解决现有技术存在仅对多操作系统中的一个操作系统的进程进行关闭导致资源释放效果不佳的问题。

[0007] 本发明的实施例根据一个方面,提供了一种多系统的资源释放方法,包括:

[0008] 主控系统监测本设备的空闲资源的数值;

[0009] 当监测到所述空闲资源的数值降低至预设的多个空闲资源数值阈值之一时,根据预设的资源释放策略,从至少两个容器系统的多个进程中,确定出当前空闲资源数据阈值对应的待关闭的进程;

[0010] 关闭确定出的待关闭的进程以释放相应的资源。

[0011] 本发明的实施例根据另一个方面,还提供了一种多系统的资源释放装置,包括:主控系统和至少两个容器系统;

[0012] 所述主控系统包括:

[0013] 空闲资源监测模块,用于监测本设备的空闲资源的数值;当监测到所述空闲资源的数值降低至预设的多个空闲资源数值阈值之一时,发送资源释放通知;

[0014] 待关闭进程确定模块,用于接收到资源释放通知后,根据预设的资源释放策略,从至少两个容器系统的多个进程中,确定出当前空闲资源数据阈值对应的待关闭的进程;

[0015] 资源释放模块,用于关闭确定出的待关闭的进程以释放相应的资源。

[0016] 较佳地,所述待关闭进程确定模块具体用于根据所述资源释放策略中多个空闲资源数据阈值与进程的综合重要度值的对应关系,确定出当前空闲资源数据阈值所对应的进程的综合重要度值,作为当前综合重要度值阈值;当所述资源释放策略中进程的综合重要度值与进程的重要度的关系为综合重要度值越大则进程的重要度越小时,从至少两个容器系统的多个进程中,确定出综合重要度值大于所述当前综合重要度值阈值的进程,作为所述当前空闲资源数据阈值对应的待关闭的进程。

[0017] 较佳地,每个容器系统包括:

[0018] 基础重要度值确定模块,用于根据所述资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值后,向所述主控系统发送;以及

[0019] 所述主控系统还包括:

[0020] 系统重要度值确定模块,用于根据所述资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出每个容器系统的系统重要度值,作为该容器系统中进程的系统重要度值;

[0021] 综合重要度值确定模块,用于对于每个容器系统中的每个进程,根据该进程的基础重要度值和系统重要度值生成该进程的综合重要度值。

[0022] 较佳地,每个容器系统包括:

[0023] 基础重要度值确定模块,用于根据所述资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值;

[0024] 系统重要度值确定模块,用于根据所述资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出该容器系统的系统重要度值,作为该容器系统中进程的系统重要度值;

[0025] 综合重要度值确定模块,用于对于该容器系统中的每个进程,根据该进程的基础重要度值和系统重要度值生成该进程的综合重要度值后,向所述主控系统发送。

[0026] 较佳地,所述综合重要度值确定模块具体用于将该进程的基础重要度值与该进程的系统重要度值进行累加,得到该进程的综合重要度值;根据该进程的系统重要度值对该进程的基础重要度值进行加权,得到该进程的综合重要度值。

[0027] 较佳地,所述资源释放模块具体用于当确定出多个待关闭的进程时,根据各待关闭的进程的综合重要度值从大到小的顺序,依次关闭待关闭的进程。

[0028] 本发明实施例中,根据预设的统一的资源释放策略统一评价多个容器系统的所有进程的重要度,从而将各容器系统的进程都纳入到统一的重要度评价体系中;无论进程属于前台的容器系统、还是后台的容器系统,都根据统一的资源释放策略,判断进程是否应当关闭;从而可以根据统一的评价结果对多个容器系统的进程进行关闭以释放所占系统资源,实现对多个容器系统中的进程所占系统资源的管控;优化了多系统下的系统资源的释

放效果,可以给用户带来便利,提升用户的体验。

[0029] 本发明附加的方面和优点将在下面的描述中部分给出,这些将从下面的描述中变得明显,或通过本发明的实践了解到。

附图说明

[0030] 本发明上述的和/或附加的方面和优点从下面结合附图对实施例的描述中将变得明显和容易理解,其中:

[0031] 图1为本发明实施例的终端设备的内部结构的框架示意图;

[0032] 图2、3分别为本发明实施例的进程的综合重要度值的两种确定方法的流程图;

[0033] 图4为本发明实施例的多系统的资源释放方法的流程图;

[0034] 图5、6分别为本发明实施例的两种多系统的资源释放装置的内部结构的框架示意图。

具体实施方式

[0035] 下面详细描述本发明的实施例,所述实施例的示例在附图中示出,其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的,仅用于解释本发明,而不能解释为对本发明的限制。

[0036] 本技术领域技术人员可以理解,除非特意声明,这里使用的单数形式“一”、“一个”、“所述”和“该”也可包括复数形式。应该进一步理解的是,本发明的说明书中使用的措辞“包括”是指存在所述特征、整数、步骤、操作、元件和/或组件,但是并不排除存在或添加一个或多个其他特征、整数、步骤、操作、元件、组件和/或它们的组。应该理解,当我们称元件被“连接”或“耦接”到另一元件时,它可以直接连接或耦接到其他元件,或者也可以存在中间元件。此外,这里使用的“连接”或“耦接”可以包括无线连接或无线耦接。这里使用的措辞“和/或”包括一个或多个相关联的列出项的全部或任一单元和全部组合。

[0037] 本技术领域技术人员可以理解,除非另外定义,这里使用的所有术语(包括技术术语和科学术语),具有与本发明所属领域中的普通技术人员的一般理解相同的意义。还应该理解的是,诸如通用字典中定义的那些术语,应该被理解为具有与现有技术的上下文中的意义一致的意义,并且除非像这里一样被特定定义,否则不会用理想化或过于正式的含义来解释。

[0038] 本技术领域技术人员可以理解,这里所使用的“终端”、“终端设备”既包括无线信号接收器的设备,其仅具备无发射能力的无线信号接收器的设备,又包括接收和发射硬件的设备,其具有能够在双向通信链路上,进行双向通信的接收和发射硬件的设备。这种设备可以包括:蜂窝或其他通信设备,其具有单线路显示器或多线路显示器或没有多线路显示器的蜂窝或其他通信设备;PCS(Personal Communications Service,个人通信系统),其可以组合语音、数据处理、传真和/或数据通信能力;PDA(Personal Digital Assistant,个人数字助理),其可以包括射频接收器、寻呼机、互联网/内联网访问、网络浏览器、记事本、日历和/或GPS(Global Positioning System,全球定位系统)接收器;常规膝上型和/或掌上型计算机或其他设备,其具有和/或包括射频接收器的常规膝上型和/或掌上型计算机或其

他设备。这里所使用的“终端”、“终端设备”可以是便携式、可运输、安装在交通工具(航空、海运和/或陆地)中的,或者适合于和/或配置为在本地运行,和/或以分布形式,运行在地球和/或空间的任何其他位置运行。这里所使用的“终端”、“终端设备”还可以是通信终端、上网终端、音乐/视频播放终端,例如可以是PDA、MID(Mobile Internet Device,移动互联网设备)和/或具有音乐/视频播放功能的移动电话,也可以是智能电视、机顶盒等设备。

[0039] 本发明实施例中,终端设备包括主控系统和至少两个容器系统。主控系统监测本设备的空闲资源的数值;当监测到空闲资源的数值降低至预设的多个空闲资源数值阈值之一时,根据预设的资源释放策略,从至少两个容器系统的多个进程中,确定出当前空闲资源数据阈值对应的待关闭的进程;关闭确定出的待关闭的进程以释放相应的资源。可见,本发明实施例中,根据预设的统一的资源释放策略,对各容器系统的每个进程是否应该释放资源进行统一评价,相当于将各容器系统的所有进程都纳入到一个统一的评价系统中,进而统一确定出当前空闲资源数据阈值下应当关闭的进程进行关闭。例如,可以关闭处于前台的容器系统的一小部分进程,同时关闭处于后台的容器系统的一大部分进程但仍保持后台的容器系统可以正常运行的状态。使得既不影响容器系统的前后台切换,又可以大幅度的释放系统资源,给用户带来便利,可以提升用户的体验。

[0040] 下面结合附图具体介绍本发明实施例的技术方案。

[0041] 本发明实施例的终端设备的内部结构的框架示意图如图1所示,包括:主控系统和两个以上容器系统。

[0042] 其中,两个以上容器系统包括第一容器系统、第二容器系统、……、第N容器系统。N为2以上的正整数。

[0043] 本发明实施例中的容器系统,是设置在以Linux container(容器)虚拟化技术创建的容器中的操作系统。操作系统可以为传统意义上的Linux操作系统或Unix操作系统,也可以是基于Linux操作系统衍生出来的Android系统或Ubuntu系统等,还可以为以Windows平台为基础的windows系统等等。实际上,本发明中的容器系统不限于前述列举的操作系统,可以涵盖所有能够在容器中运行的操作系统。为便于描述,下面以Android系统作为容器系统为例阐述本发明的技术方案。

[0044] 主控系统可以是上述传统的操作系统,也可以是对传统的操作系统进行改进得到的操作系统。内核为kernel,或者在kernel基础上增加功能模块之后得到的增强型kernel。较佳地,主控系统可以包括内核;上述容器系统通过主控系统调用或访问内核以实现各种功能。或者,容器系统和主控系统共享内核;本发明的容器系统和主控系统包含现有操作系统中kernel之外的部分,例如框架层和应用层的部分;容器系统和主控系统都可以独立调用或访问内核以实现各种功能。

[0045] 主控系统主要用于对多个容器系统进行前后台管理,与各容器系统进行交互等。

[0046] 较佳地,主控系统可以通过容器通道与容器系统进行通信。进一步,容器通道可以是socket(套接字)通道。

[0047] 容器系统向主控系统发送容器通道创建请求,该容器通道创建请求中包括容器通道对应的文件描述符、文件权限、允许访问该容器通道的两个以上操作系统的系统标识、允许的最大访问数量、以及待生成文件的描述符的标识等。两个以上操作系统包括(发送容器通道创建请求的)容器系统、和该容器系统期待交互的操作系统。

[0048] 主控系统接到该容器系统的容器通道创建请求后,创建一对socket文件描述符(socketpair)和连接该对文件描述符的容器通道,并为该对文件描述符命名。将容器通道创建请求中的文件权限、允许访问的系统标识、允许的最大访问数量、以及待生成文件的描述符的标识记录到所创建的容器通道中。

[0049] 主控系统将容器通道及其标识和一对文件描述符、以及允许访问该容器通道的系统标识注册在内核的NSS(Name Space Server,域名空间服务)虚拟设备中。主控系统和内核都对对应记录已经注册的容器通道及其标识和文件描述符对、以及允许访问该容器通道的系统标识。

[0050] 内核的NSS将容器通道的标识、以及允许访问该容器通道的各系统标识返回给该容器系统。并将容器通道的标识和允许访问该容器通道的各系统标识,分别向与各系统标识相符的操作系统返回。

[0051] 该容器系统根据获取的容器通道的标识从NSS中获取该容器通道的文件描述符对中的一个文件描述符。

[0052] 主控系统接收到主控系统针对上述容器通道的获取请求后,查找出已经注册的该容器通道的标识,向该主控系统返回。

[0053] 主控系统获取容器通道的标识后,向内核中的NSS虚拟设备发送,携带有该容器通道的标识的文件描述符获取请求。

[0054] 内核中的NSS虚拟设备根据接收到的该文件描述符获取请求中的容器通道的标识,从已注册的容器通道及其标识和文件描述符对中,查找出该容器通道的文件描述符对中的另一个文件描述符返回。较佳地,NSS校验发送文件描述符获取请求的容器系统的系统标识、访问数量等安全信息是否与容器通道中记录的对应信息相符;若校验结果为相符,则根据接收到的该文件描述符获取请求中的容器通道的标识,从已注册的容器通道及其标识和文件描述符对中,查找出该容器通道的文件描述符对中的另一个文件描述符,向主控系统返回。

[0055] 相当于,将连接容器通道两端的一对文件描述符中的一个文件描述符向容器系统返回,将另一个文件描述符向主控系统的进程返回。

[0056] 主控系统获取内核返回的另一个文件描述符后,主控系统与容器系统之间的容器通道建立完毕。主控系统可以通过该另一个文件描述符,与拥有该另一个文件描述符所属容器通道的一个文件描述符的容器系统进行通信。

[0057] 较佳地,主控系统对于一对文件描述符,根据返回给一个容器系统的一个文件描述符,从内核获取socket描述文件的结构体,与该一个文件描述符相关联,并记录在容器通道中。之后,将容器通道注册到内核的NSS虚拟设备中。

[0058] 内核中的NSS虚拟设备接收到主控系统的进程的文件描述符获取请求后,查询出主控系统的进程中未使用的文件描述符,将该未使用的文件描述符与容器通道中保存的socket描述文件的文件结构体相关联,并向主控系统的该进程返回关联后的文件描述符。此时,关联后的文件描述符相当于同一容器通道的文件描述符中的另一个文件描述符。

[0059] 本发明实施例中,终端设备中预设资源释放策略。资源释放策略包括:多个空闲资源数据阈值与进程的综合重要度值的对应关系、进程的综合重要度值与进程的重要度的关系、进程的类型与基础重要度值的对应关系、以及容器系统的前后台状态与系统重要度

值的对应关系。

[0060] 其中,进程的综合重要度值与进程的重要度的关系,可以是进程的综合重要度值越大则进程的重要度越小、进程的综合重要度值越小则进程的重要度越小、或者其它进程的综合重要度值与重要度之间的关系。

[0061] 较佳地,为了便于描述和理解,本文后续以进程的综合重要度值越大则进程的重要度越小为例,介绍本发明实施例的技术方案。可以理解,对于同一容器系统中的进程,当进程的综合重要度值越大则进程的重要度越小时,进程的基础重要度值越大则进程的重要度越小。

[0062] 至于资源释放策略中的其它内容,将在后续详述,此处不再赘述。

[0063] 较佳地,本发明实施例中的资源释放策略可以存储于主控系统中,例如存储于主控系统的内核中。当然,容器系统可以通过主控系统共享内核。对于每个容器系统,当该容器系统被主控系统启动后,主控系统可以通过容器通道向该容器系统推送资源释放策略;也可以由该容器系统通过容器通道从主控系统中获取资源释放策略。

[0064] 较佳地,每个容器系统可以配置有涉及资源释放策略的人机交互界面;支持用户通过容器系统的人机交互界面,输入针对资源释放策略的调整内容,由主控系统根据调整内容对资源释放策略进行相应更新并存储。

[0065] 本发明实施例中提供了一种进程的综合重要度值的确定方法,该方法的流程示意图如图2所示,包括下述步骤:

[0066] S201:每个容器系统根据资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值后,向主控系统发送。

[0067] 本发明实施例中,资源释放策略中进程的类型与基础重要度值的对应关系,包括:依照基础重要度值从大到小的顺序,与基础重要度值对应的进程类型分别为:空闲进程、后台隐藏应用进程、后台服务、桌面应用进程、服务、备份应用进程、高负载应用进程、可感知应用进程、可见应用进程、前台应用进程、驻留程序进程、系统进程。其中,同一容器系统中进程的基础重要度值越大,则进程的重要度越小。本领域技术人员可以根据经验数据、实验数据、历史数据和/或实际情况,预先设定多种进程的类型各自对应的基础重要度值,并记载到资源释放策略的进程的类型与基础重要度值的对应关系中。

[0068] 其中,前台 (foreground) 进程可以是目前正在屏幕上显示的进程和一些系统进程。例如,Dialer Storage,Google Search等系统进程就是前台进程;再如,当运行一个浏览器,当该浏览器界面在前台显示时,浏览器属于前台进程 (foreground),但一旦接收到用户针对home键的选择操作则将屏幕切换回到主界面,此时浏览器就变成了后台程序 (background)。

[0069] 可见 (visible) 应用进程是一些不在前台,但用户依然可见的进程。例如,时钟、天气或新闻等widget应用,以及输入法应用等,都属于visible应用进程。这部分进程虽然不在前台,但与用户的使用也密切相关。因此一旦时钟、天气或新闻等widget被关闭,它们将无法同步,导致用户无法获取最新的时钟、天气或新闻,容易降低用户的体验。若输入法应用被关闭,导致用户输入时需要重新启动输入法应用。

[0070] 可感知应用 (PERCEPTIBLEAPP) 是指在终端设备的屏幕上不直接显示、但可以被用户感知的应用。较佳地,可感知应用可以通过输出声音、和/或振动等非显示信息被用户感

知。例如，正在后台进行音乐播放的音乐播放器应用。

[0071] 高负载应用 (HEAVYWEIGHTAPP) 是指占用终端设备的诸如处理器和/或内存等系统资源的比例超过预设的资源比例阈值的应用。技术人员可以根据系统默认数据、历史数据、实验数据、经验数据和/或实际情况，预先设定资源比例阈值。例如，将系统资源中的处理器资源比例阈值预设为70%；当一个实时视频会议应用占用处理器资源的比例、超过处理器资源的70%时，确定该实时视频会议应用占用处理器资源的比例超过处理器资源比例阈值，将该实时视频会议应用确定为高负载应用。

[0072] 服务 (server) 可以是目前正在运行的一些服务，和通常不会被系统自带的进程管理器终止关闭的服务。这些服务通常包括：通信相关服务、企业套件相关服务、电子邮件相关服务和联系人相关服务等。例如：拨号服务、谷歌企业套件、Gmail (Google的免费网络邮件服务) 内部存储、或联系人内部存储等服务。这部分服务虽然属于服务，但很大一部分与系统功能依然息息相关，需要这些服务处于运行状态以支持用户时常使用的相关系统功能。

[0073] 桌面应用 (home app) 进程，即launcher进程，其主要用于保证在多任务切换之后，可以快速返回到home界面而不需重新加载launcher。

[0074] 后台隐藏 (hidden) 应用进程：虽然用了hidden这个词，但实际即是后台进程 (background)，就是通常意义上启动后被切换到后台的进程，如浏览器，阅读器等。当程序显示在屏幕上时，该程序所运行的进程即为前台 (foreground) 进程，一旦接收到用户针对home键的选择操作后将屏幕返回至主界面 (注意是对home键而不是back键的选择操作)，该程序的进程就驻留在后台，成为后台隐藏应用进程。

[0075] 内容供应 (content provider) 节点：没有程序实体，仅提供内容供其它程序调用，例如日历供应节点，邮件供应节点等。

[0076] 空 (empty) 进程：没有任何东西在内运行的进程。有些程序，比如BTE (Better Terminal Emulator, 超级终端模拟器)，在程序退出后，依然会在系统中驻留一个空进程，这个进程里没有任何数据在运行，作用往往是提高该程序下次的启动速度或者记录程序的一些历史信息。这部分进程是应该最先终止关闭的。

[0077] 终端设备中的每个容器系统，确定出该容器系统中每个进程的类型；根据资源释放策略中进程的类型与基础重要度值的对应关系，确定出该容器系统中每个进程的类型所对应的的基础重要度值。

[0078] 其中，进程的基础重要度值，用以表征该进程在其所属操作系统 (例如容器系统) 范围内的重要程度；而进程的综合重要度值，用以表征该进程在整个多操作系统范围内的重要程度，即除了需要考虑该进程在其所属操作系统中的重要程度之外，还需要考虑其所属操作系统在各操作系统中的重要程度。

[0079] 例如，某一容器系统确定出其中各类型的进程的如下基础重要度值：

[0080] `HIDDEN_APP_MAX_ADJ=15;`

[0081] `HIDDEN_APP_MIN_ADJ=9;`

[0082] `SERVICE_B_ADJ=8;`

[0083] `PREVIOUS_APP_ADJ=7;`

[0084] `HOME_APP_ADJ=6;`

[0085] SERVICE_ADJ=5;
[0086] BACKUP_APP_ADJ=4;
[0087] HEAVY_WEIGHT_APP_ADJ=3;
[0088] PERCEPTIBLE_APP_ADJ=2;
[0089] VISIBLE_APP_ADJ=1;
[0090] FOREGROUND_APP_ADJ=0;
[0091] PERSISTENT_PROC_ADJ=-12;
[0092] SYSTEM_ADJ=-16。

[0093] 其中,ADJ表示进程的基础重要度值,-16表示负16;SERVICE_B(Service Background,后台服务)、PREVIOUS_APP、HOME_APP、SERVICE、BACKUP_APP、HEAVY_WEIGHT_APP、PERCEPTIBLE_APP、VISIBLE_APP、FOREGROUND_APP、PERSISTENT_PROC、SYSTEM分别表示后台服务、上一次应用进程、桌面应用进程、服务、备份应用进程、高负载应用进程、可感知应用进程、可见应用进程、前台应用进程、驻留程序进程、系统进程。HIDDEN_APP_MAX_ADJ和HIDDEN_APP_MIN_ADJ分别表示隐藏应用进程的空闲进程的最大和最小基础重要度值;隐藏应用进程,包括上述后台隐藏应用进程、内容供应节点和空进程,其中空进程的基础重要度值即为HIDDEN_APP_MAX_ADJ。

[0094] 可以理解,根据前述进程的综合重要度值越大则进程的重要度越小的情况下,同一容器系统中,进程的基础重要度值越大则进程的重要度越小。

[0095] 容器系统确定出该容器系统中每个进程的基础重要度值后,将各进程的进程标识和基础重要度值,通过容器通道向主控系统发送。

[0096] S202:主控系统根据资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出每个容器系统的系统重要度值,作为该容器系统中进程的系统重要度值。

[0097] 主控系统确定出每个容器系统的当前状态,例如容器系统的前后台状态;根据资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,分别确定出处于前台、后台的容器系统的系统重要度值;将每个容器系统的系统重要度值,作为该容器系统中进程的系统重要度值。其中,前后台状态与系统重要度值的对应关系包括:处于后台的容器系统的系统重要度值大于处于前台的容器系统的系统重要度值、以及两者之间的差值。

[0098] 例如,资源释放策略中容器系统的前后台状态与系统重要度值的对应关系具体为:处于后台的容器系统的系统重要度值,比处于前台的容器系统的系统重要度值大2;则可以将处于前台的容器系统的系统重要度值设置为0,将处于后台的容器系统的系统重要度值设置为2。

[0099] 可以理解,根据前述进程的综合重要度值越大则进程的重要度越小的情况下,操作系统的系统重要度值越大则该操作系统中进程的重要度越小。

[0100] 较佳地,主控系统可以根据与上述步骤中容器系统确定其进程的基础重要性值相同的方法,确定出主控系统中每个进程的基础重要性值。

[0101] 进一步,为了保证主控系统在所有容器系统都关闭后仍能正常运行,可以将主控系统中的每个进程的基础重要度值,设定为不大于处于前台的容器系统的前台程序进程的基础重要度值。例如,将主控系统中的每个进程的基础重要度值都设置为负值。

[0102] 较佳地,资源释放策略中容器系统的前后台状态与系统重要度值的对应关系中可

以包括:主控系统的系统重要度值不大于处于前台的容器系统的系统重要度值。例如,可以将主控系统的系统重要度值设置为0或负值。

[0103] S203:主控系统对于每个容器系统中的每个进程,根据该进程的基础重要度值和系统重要度值生成该进程的综合重要度值。

[0104] 主控系统对于每个容器系统中的每个进程,根据该进程的基础重要度值和系统重要度值,将该进程的基础重要度值与该进程的系统重要度值进行累加,得到该进程的综合重要度值。

[0105] 例如,当处于前台的容器系统中各类型的进程的基础重要度值如上述步骤S201中的例子所示、且该处于前台的容器系统的系统重要度值为0时,确定出该处于前台的容器系统的各类型的进程的如下所述的综合重要度值:

[0106] HIDDEN_APP_MAX_CADJ=15;

[0107] HIDDEN_APP_MIN_CADJ=9;

[0108] SERVICE_B_CADJ=8;

[0109] PREVIOUS_APP_CADJ=7;

[0110] HOME_APP_CADJ=6;

[0111] SERVICE_CADJ=5;

[0112] BACKUP_APP_CADJ=4;

[0113] HEAVY_WEIGHT_APP_CADJ=3;

[0114] PERCEPTIBLE_APP_CADJ=2;

[0115] VISIBLE_APP_CADJ=1;

[0116] FOREGROUND_APP_CADJ=0;

[0117] PERSISTENT_PROC_CADJ=-12;

[0118] SYSTEM_CADJ=-16。

[0119] 其中,C(Compositive,综合的)ADJ表示进程的综合重要度值。

[0120] 而且,当处于后台的容器系统中各类型的进程的基础重要度值如上述步骤S201中的例子所示、且该处于后台的容器系统的系统重要度值为2时,确定出该处于前台的容器系统的各类型的进程的如下所述的综合重要度值:

[0121] HIDDEN_APP_MAX_CADJ=17;

[0122] HIDDEN_APP_MIN_CADJ=11;

[0123] SERVICE_B_CADJ=10;

[0124] PREVIOUS_APP_CADJ=9;

[0125] HOME_APP_CADJ=8;

[0126] SERVICE_CADJ=7;

[0127] BACKUP_APP_CADJ=6;

[0128] HEAVY_WEIGHT_APP_CADJ=5;

[0129] PERCEPTIBLE_APP_CADJ=4;

[0130] VISIBLE_APP_CADJ=3;

[0131] FOREGROUND_APP_CADJ=2;

[0132] PERSISTENT_PROC_CADJ=-10;

[0133] SYSTEM_CADJ=-14。

[0134] 容易看出,本步骤的上述例子中,处于后台的容器系统的一些进程的综合重要度值,大于处于前台的容器系统的一些进程的综合重要度值;同一类型的应用进程,在处于后台的容器系统中的综合重要度值,大于其在处于前台的容器系统中的综合重要度值。

[0135] 例如,处于后台的容器系统的前台应用进程FOREGROUND_APP的综合重要度值为2,大小处于前台的容器系统的可见应用进程VISIBLE_APP的综合重要度值为1。处于后台的容器系统的前台应用进程FOREGROUND_APP的重要程度不如处于前台的容器系统的可见应用进程VISIBLE_APP。

[0136] 再如,对于同一类型的前台应用进程FOREGROUND_APP,其在处于后台的容器系统中的综合重要度值为2,在处于前台的容器系统中的综合重要度值为0。

[0137] 主控系统对于每个容器系统中的每个进程,根据该进程的基础重要度值和系统重要度值,根据该进程的系统重要度值对该进程的基础重要度值进行加权,得到该进程的综合重要度值。

[0138] 较佳地,主控系统对于主控系统的每个进程,根据上述步骤中确定出的该进程的基础重要度值和系统重要度值,以本步骤中上述同样的方法确定出该进程的综合重要度值。

[0139] 本发明实施例中提供了另一种进程的综合重要度值的确定方法,该方法的流程示意图如图3所示,包括下述步骤:

[0140] S301:每个容器系统根据资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值。

[0141] 本步骤中,每个容器系统确定出该容器系统中每个进程的基础重要度值的方法,与上述步骤S201中每个容器系统确定出该容器系统中每个进程的基础重要度值的具体方法一致,此处不再赘述。

[0142] S302:每个容器系统根据资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出该容器系统的系统重要度值,作为该容器系统中进程的系统重要度值。

[0143] 每个容器系统可以从该容器系统中获取预存的资源释放策略,并检测出该容器系统的前后台状态。

[0144] 或者,每个容器系统可以通过各自的容器通道从主控系统中获取该容器系统的前后台状态以及资源释放策略。

[0145] 每个容器系统根据该容器系统的前后台状态、以及资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出该容器系统的系统重要度值;将该容器系统的系统重要度值,作为该容器系统中进程的系统重要度值。其中,前后台状态与系统重要度值的对应关系包括:处于前台的容器系统的系统重要度值小于处于后台的容器系统的系统重要度值、以及两者之间的差值。

[0146] 例如,一个容器系统的处于前台状态,且资源释放策略中容器系统的前后台状态与系统重要度值的对应关系具体为:处于后台的容器系统的系统重要度值,比处于前台的容器系统的系统重要度值大2;则可以将该处于前台的容器系统的系统重要度值设置为0。

[0147] 可以理解,根据前述进程的综合重要度值越大则进程的重要度越小的情况下,操作系统的系统重要度值越大则该操作系统中进程的重要度越小。

[0148] S303:每个容器系统对于其中的每个进程,根据该进程的基础重要度值和系统重要度值生成该进程的综合重要度值后,向主控系统发送。

[0149] 每个容器系统对于其中的每个进程,根据该进程的基础重要度值和系统重要度值,将该进程的基础重要度值与该进程的系统重要度值进行累加,得到该进程的综合重要度值。

[0150] 例如,当一个容器系统处于后台状态时,当处于后台的容器系统中各类型的进程的基础重要度值如上述步骤S201中的例子所示、且该处于后台的容器系统的系统重要度值为2时,确定出该处于前台的容器系统的各类型的进程的如下所述的综合重要度值:

[0151] HIDDEN_APP_MAX_CADJ=17;

[0152] HIDDEN_APP_MIN_CADJ=11;

[0153] SERVICE_B_CADJ=10;

[0154] PREVIOUS_APP_CADJ=9;

[0155] HOME_APP_CADJ=8;

[0156] SERVICE_CADJ=7;

[0157] BACKUP_APP_CADJ=6;

[0158] HEAVY_WEIGHT_APP_CADJ=5;

[0159] PERCEPTIBLE_APP_CADJ=4;

[0160] VISIBLE_APP_CADJ=3;

[0161] FOREGROUND_APP_CADJ=2;

[0162] PERSISTENT_PROC_CADJ=-10;

[0163] SYSTEM_CADJ=-14。

[0164] 每个容器系统对于其中的每个进程,根据该进程的基础重要度值和系统重要度值,根据该进程的系统重要度值对该进程的基础重要度值进行加权,得到该进程的综合重要度值。

[0165] 每个容器系统确定出各自的每个进程的综合重要度值后,将该进程的进程标识和综合重要度值通过各自的容器通道向主控系统发送。

[0166] 较佳地,主控系统可以根据与上述步骤中容器系统确定其进程的基础重要性值相同的方法,确定出主控系统中每个进程的基础重要性值。

[0167] 进一步,为了保证主控系统在所有容器系统都关闭后仍能正常运行,可以将主控系统中的每个进程的基础重要度值,设定为不大于处于前台的容器系统的前台程序进程的基础重要度值。例如,将将主控系统中的每个进程的基础重要度值都设置为负值。

[0168] 较佳地,资源释放策略中容器系统的前后台状态与系统重要度值的对应关系中包括:主控系统的系统重要度值不大于处于前台的容器系统的系统重要度值。例如,可以将主控系统的系统重要度值设置为0或负值。

[0169] 较佳地,主控系统对于主控系统的每个进程,根据确定出的该进程的基础重要度值和系统重要度值,以本步骤中上述同样的方法确定出该进程的综合重要度值。

[0170] 更优的,上述图2-3所示的方法中,给出了处于前台的容器系统、以及处于后台的容器系统中的进程的综合重要度值的确定方法。在实际操作中,当容器系统发生前后台切换后,可以利用同样的方法重新确定各容器系统的进程的综合重要度值。

[0171] 下面介绍本发明实施例提供的多系统的资源释放方法,该方法的流程示意图如图4所示,包括下述步骤:

[0172] S401:主控系统监测本设备的空闲资源的数值。

[0173] 主控系统监测本终端设备的空闲系统资源的数值。例如,监测本终端设备中空闲内存的数值。

[0174] S402:当监测到空闲资源的数值降低至预设的多个空闲资源数值阈值之一时,根据预设的资源释放策略,从至少两个容器系统的多个进程中,确定出当前空闲资源数据阈值对应的待关闭的进程。

[0175] 资源释放策略中多个空闲资源数据阈值与进程的综合重要度值的对应关系,可以包括:多个空闲资源数据阈值与多个进程的综合重要度值的对应关系。例如,空闲内存数据阈值20MB(Mega Byte,兆字节)与进程的综合重要度值0相对应,表明当空闲内存降低到空闲内存数据阈值20MB,应当将综合重要度值不小于0的进程都关闭;空闲内存数据阈值60MB与进程的综合重要度值4相对应;空闲内存数据阈值100MB与进程的综合重要度值8相对应。

[0176] 当主控系统监测到空闲资源的数值降低至预设的多个空闲资源数值阈值之一时,根据资源释放策略中多个空闲资源数据阈值与进程的综合重要度值的对应关系,确定出当前空闲资源数据阈值所对应的进程的综合重要度值,作为当前综合重要度值阈值。

[0177] 主控系统从至少两个容器系统的多个进程中,确定出综合重要度值大于当前综合重要度值阈值的进程,作为当前空闲资源数据阈值对应的待关闭的进程。

[0178] 例如,当主控系统监测到当前的空闲资源的数值,降低至预设的多个空闲资源数值阈值之一60MB时,对应的进程的综合重要度值为4;确定出处于前台的容器系统中的空闲进程、后台隐藏应用进程、后台服务、桌面应用进程、服务、备份应用进程,以及处于后台的容器系统中的空闲进程、后台隐藏应用进程、后台服务、桌面应用进程、服务、备份应用进程、高负载应用进程、可感知应用进程,都为待关闭的进程。

[0179] 较佳地,主控系统从本主控系统和至少两个容器系统的多个进程中,确定出综合重要度值大于当前综合重要度值阈值的进程,作为当前空闲资源数据阈值对应的待关闭的进程。

[0180] S403:关闭确定出的待关闭的进程以释放相应的资源。

[0181] 主控系统对上述步骤确定出的待关闭的进程进行关闭,从而释放出被关闭的进程所占用的系统资源。

[0182] 事实上,根据本发明实施例中如上述图1-4所示的技术方案,对于多个容器系统以及主控系统的所有进程,都根据进程的综合重要度值统一评价这些进程的重要度,从而将各容器系统和主控系统的进程都纳入到统一的重要度评价体系中;无论进程属于前台的容器系统、后台的容器系统还是主控系统,都根据统一的综合重要度值,判断进程是否应当关闭;从而可以对多个容器系统的进程进行关闭以释放所占系统资源,实现对多个容器系统中的进程所占系统资源的管控;优化了多系统下的系统资源的释放效果,可以给用户带来便利,提升用户的体验。

[0183] 而且,本发明的发明人发现,终端设备得到多个容器系统和主控系统的进程,实际上通常都运行于该终端设备的内存中;利用本发明实施例,由于具有基于综合重要度值的统一的重要度评价系统,无需对进程所属的操作系统进行识别,直接根据内存中进程的综

合重要值判断该进程是否需要关闭即可;节省了识别并判断进程所属操作系统的步骤,提升了释放进程所占系统资源的效率。

[0184] 进一步,在进程的综合重要度值越大则进程的重要度越小的情况下,上述主控系统中的进程、容器系统中的系统进程和驻留程序进程的综合重要度值,明显比多个容器系统的其他进程的综合重要度值低一大截;合理设置资源释放策略中的内容,可以使得即使多个容器系统的其它进程全部被关闭后,主控系统中的进程、容器系统中的系统进程和驻留程序进程依然得以保留,从而可以保证主控系统和容器系统得以正常运行而不崩溃,使得容器系统之间依然可以随时进行前后台切换,并随时可以添加运行容器系统的进程;从而可以在支持用户对容器系统进行前后台切换的情况下,管控各容器系统的进程所占系统资源的释放,扩展了系统资源释放的范围和程度,给用户带来便利,提升用户的体验。

[0185] 基于上述多系统的资源释放方法,本发明实施例中还提供了一种多系统的资源释放装置,该装置设置于本发明实施例的终端设备中,该装置的内部结构的框架示意图如图5所示,包括:主控系统和至少两个容器系统。

[0186] 其中,主控系统包括:空闲资源监测模块501、待关闭进程确定模块502和资源释放模块503。

[0187] 空闲资源监测模块501用于监测本设备的空闲资源的数值;当监测到空闲资源的数值降低至预设的多个空闲资源数值阈值之一时,发送资源释放通知。

[0188] 待关闭进程确定模块502用于接收到空闲资源监测模块501发送的资源释放通知后,根据预设的资源释放策略,从至少两个容器系统的多个进程中,确定出当前空闲资源数据阈值对应的待关闭的进程。

[0189] 资源释放模块503用于关闭待关闭进程确定模块502确定出的待关闭的进程以释放相应的资源。

[0190] 较佳地,待关闭进程确定模块502具体用于根据资源释放策略中多个空闲资源数据阈值与进程的综合重要度值的对应关系,确定出当前空闲资源数据阈值所对应的进程的综合重要度值,作为当前综合重要度值阈值;当资源释放策略中进程的综合重要度值与进程的重要度的关系为综合重要度值越大则进程的重要度越小时,从至少两个容器系统的多个进程中,确定出综合重要度值大于当前综合重要度值阈值的进程,作为当前空闲资源数据阈值对应的待关闭的进程。

[0191] 较佳地,如图5所示,本发明实施例的一种多系统的资源释放装置中的每个容器系统包括:基础重要度值确定模块511。

[0192] 基础重要度值确定模块511用于根据资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值后,向主控系统发送。

[0193] 以及,如图5所示,本发明实施例的一种多系统的资源释放装置中的主控系统还包括:系统重要度值确定模块504和综合重要度值确定模块505。

[0194] 其中,系统重要度值确定模块504用于根据资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出每个容器系统的系统重要度值,作为该容器系统中进程的系统重要度值。

[0195] 综合重要度值确定模块505用于对于每个容器系统中的每个进程,根据基础重要度值确定模块511发送的该进程的基础重要度值和系统重要度值确定模块504确定出的系

统重要度值,生成该进程的综合重要度值。

[0196] 进一步,综合重要度值确定模块505具体用于将该进程的基础重要度值与该进程的系统重要度值进行累加,得到该进程的综合重要度值;根据该进程的系统重要度值对该进程的基础重要度值进行加权,得到该进程的综合重要度值。

[0197] 较佳地,资源释放模块503具体用于当待关闭进程确定模块502确定出多个待关闭的进程时,根据各待关闭的进程的综合重要度值从大到小的顺序,依次关闭待关闭的进程。

[0198] 上述主控系统中的空闲资源监测模块501、待关闭进程确定模块502、资源释放模块503、系统重要度值确定模块504和综合重要度值确定模块505,以及容器系统中的基础重要度值确定模块511功能的实现方法,可以参考上述如图1-4所示的框架结构和流程步骤的具体内容,此处不再赘述。

[0199] 基于上述多系统的资源释放方法,本发明实施例中还提供了另一种多系统的资源释放装置,该装置设置于本发明实施例的终端设备中,该装置的内部结构的框架示意图如图6所示,包括:主控系统和至少两个容器系统。

[0200] 其中,主控系统包括:空闲资源监测模块601、待关闭进程确定模块602和资源释放模块603。

[0201] 空闲资源监测模块601用于监测本设备的空闲资源的数值;当监测到空闲资源的数值降低至预设的多个空闲资源数值阈值之一时,发送资源释放通知。

[0202] 待关闭进程确定模块602用于接收到空闲资源监测模块601发送的资源释放通知后,根据预设的资源释放策略,从至少两个容器系统的多个进程中,确定出当前空闲资源数据阈值对应的待关闭的进程。

[0203] 资源释放模块603用于关闭待关闭进程确定模块602确定出的待关闭的进程以释放相应的资源。

[0204] 较佳地,待关闭进程确定模块602具体用于根据资源释放策略中多个空闲资源数据阈值与进程的综合重要度值的对应关系,确定出当前空闲资源数据阈值所对应的进程的综合重要度值,作为当前综合重要度值阈值;当资源释放策略中进程的综合重要度值与进程的重要度的关系为综合重要度值越大则进程的重要度越小时,从至少两个容器系统的多个进程中,确定出综合重要度值大于当前综合重要度值阈值的进程,作为当前空闲资源数据阈值对应的待关闭的进程。

[0205] 较佳地,如图6所示,本发明实施例的另一种多系统的资源释放装置中的每个容器系统包括:基础重要度值确定模块611、系统重要度值确定模块612和综合重要度值确定模块613。

[0206] 基础重要度值确定模块611用于根据资源释放策略中进程的类型与基础重要度值的对应关系,确定出该容器系统中每个进程的基础重要度值。

[0207] 系统重要度值确定模块612用于根据资源释放策略中容器系统的前后台状态与系统重要度值的对应关系,确定出该容器系统的系统重要度值,作为该容器系统中进程的系统重要度值。

[0208] 综合重要度值确定模块613用于对于该容器系统中的每个进程,根据基础重要度值确定模块611确定出的该进程的基础重要度值和系统重要度值确定模块612确定出的系统重要度值生成该进程的综合重要度值后,向主控系统中的待关闭进程确定模块602发送。

[0209] 进一步,综合重要度值确定模块613具体用于将该进程的基础重要度值与该进程的系统重要度值进行累加,得到该进程的综合重要度值;根据该进程的系统重要度值对该进程的基础重要度值进行加权,得到该进程的综合重要度值。

[0210] 较佳地,资源释放模块603具体用于当待关闭进程确定模块602确定出多个待关闭的进程时,根据各待关闭的进程的综合重要度值从大到小的顺序,依次关闭待关闭的进程。

[0211] 上述主控系统中的空闲资源监测模块601、待关闭进程确定模块602、资源释放模块603,以及容器系统中的基础重要度值确定模块611、系统重要度值确定模块612和综合重要度值确定模块613功能的实现方法,可以参考上述如图1-4所示的框架结构和流程步骤的具体内容,此处不再赘述。

[0212] 本发明实施例中,上述图5所示装置中的空闲资源监测模块501、待关闭进程确定模块502、资源释放模块503和基础重要度值确定模块511,分别与上述图6所示装置中的空闲资源监测模块601、待关闭进程确定模块602、资源释放模块603和基础重要度值确定模块611的功能相一致。系统重要度值确定模块504和综合重要度值确定模块505,与系统重要度值确定模块612和综合重要度值确定模块613的主要功能实际上是一致的,模块的不同标号表示该模块属于不同的操作系统。

[0213] 本发明实施例中,根据预设的统一的资源释放策略统一评价多个容器系统的所有进程的重要度,从而将各容器系统的进程都纳入到统一的重要度评价体系中;无论进程属于前台的容器系统、还是后台的容器系统,都根据统一的资源释放策略,判断进程是否应当关闭;从而可以根据统一的评价结果对多个容器系统的进程进行关闭以释放所占系统资源,实现对多个容器系统中的进程所占系统资源的管控;优化了多系统下的系统资源的释放效果,可以给用户带来便利,提升用户的体验。

[0214] 而且,本发明实施例中,终端设备得到多个容器系统和主控系统的进程,实际上通常都运行于该终端设备的内存中;利用本发明实施例,由于具有基于资源释放策略的统一的重要度评价系统,无需对进程所属的操作系统进行识别,直接根据资源释放策略判断该进程是否需要关闭即可;节省了识别并判断进程所属操作系统的步骤,提升了释放进程所占系统资源的效率。

[0215] 进一步,本发明实施例中,在资源释放策略中的进程的综合重要度值越大则进程的重要度越小的情况下,主控系统中的进程、容器系统中的系统进程和驻留程序进程的综合重要度值,明显小于多个容器系统的其他进程的综合重要度值;合理设置资源释放策略中各种类型的进程的综合重要度值等内容,可以使得即使多个容器系统的其它进程全部被关闭后,主控系统中的进程、容器系统中的系统进程和驻留程序进程依然得以保留,从而可以保证主控系统和容器系统得以正常运行而不崩溃,使得容器系统之间依然可以随时进行前后台切换,并随时可以添加运行容器系统的进程;从而可以在支持用户对容器系统进行前后台切换的情况下,管控各容器系统的进程所占系统资源的释放,扩展了系统资源释放的范围和程度,给用户带来便利,提升用户的体验。

[0216] 此外,本发明实施例中,当确定出多个待关闭的进程时,根据各待关闭的进程的综合重要度值从大到小的顺序,依次关闭待关闭的进程。由于进程的综合重要度值实际上是从影响用户使用的角度进行设置的,因此对于用户而言,利用本发明实施例可以使得较为不重要的进程关闭在先,较为重要的进程关闭再后;尽可能地为用户考虑,给用户带来便

利,从而进一步提升用户的体验。

[0217] 本技术领域技术人员可以理解,本发明包括涉及用于执行本申请中所述操作中的一项或多项的设备。这些设备可以为所需的目的而专门设计和制造,或者也可以包括通用计算机中的已知设备。这些设备具有存储在其内的计算机程序,这些计算机程序选择性地激活或重构。这样的计算机程序可以被存储在设备(例如,计算机)可读介质中或者存储在适于存储电子指令并分别耦联到总线的任何类型的介质中,所述计算机可读介质包括但不限于任何类型的盘(包括软盘、硬盘、光盘、CD-ROM、和磁光盘)、ROM(Read-Only Memory,只读存储器)、RAM(Random Access Memory,随即存储器)、EPROM(Erasable Programmable Read-Only Memory,可擦写可编程只读存储器)、EEPROM(Electrically Erasable Programmable Read-Only Memory,电可擦可编程只读存储器)、闪存、磁性卡片或光线卡片。也就是,可读介质包括由设备(例如,计算机)以能够读的形式存储或传输信息的任何介质。

[0218] 本技术领域技术人员可以理解,可以用计算机程序指令来实现这些结构图和/或框图和/或流图中的每个框以及这些结构图和/或框图和/或流图中的框的组合。本技术领域技术人员可以理解,可以将这些计算机程序指令提供给通用计算机、专业计算机或其他可编程数据处理方法的处理器来实现,从而通过计算机或其他可编程数据处理方法的处理器来执行本发明公开的结构图和/或框图和/或流图的框或多个框中指定的方案。

[0219] 本技术领域技术人员可以理解,本发明中已经讨论过的各种操作、方法、流程中的步骤、措施、方案可以被交替、更改、组合或删除。进一步地,具有本发明中已经讨论过的各种操作、方法、流程中的其他步骤、措施、方案也可以被交替、更改、重排、分解、组合或删除。进一步地,现有技术中的具有与本发明中公开的各种操作、方法、流程中的步骤、措施、方案也可以被交替、更改、重排、分解、组合或删除。

[0220] 以上所述仅是本发明的部分实施方式,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。

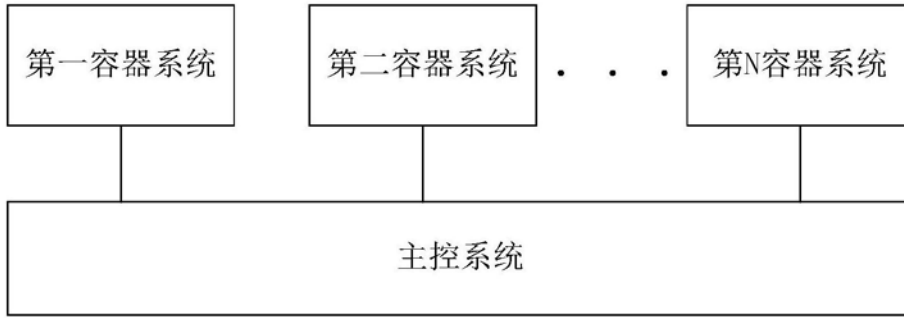


图1

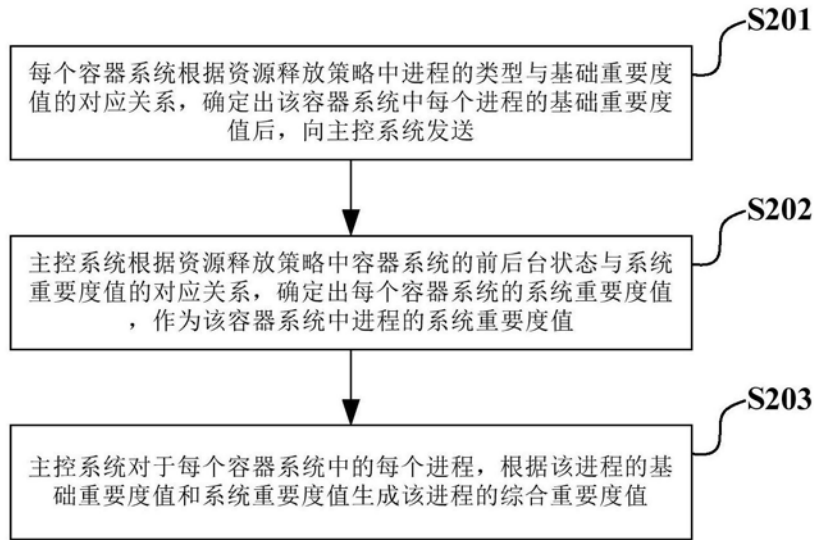


图2

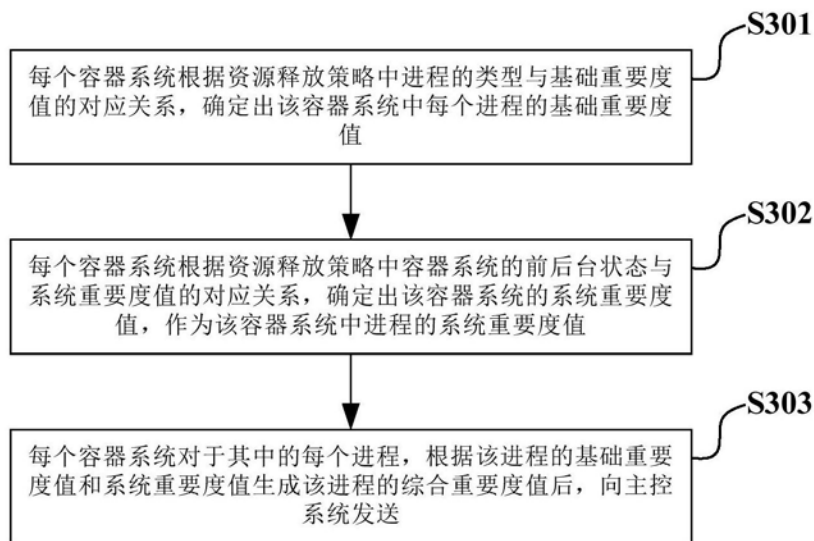


图3

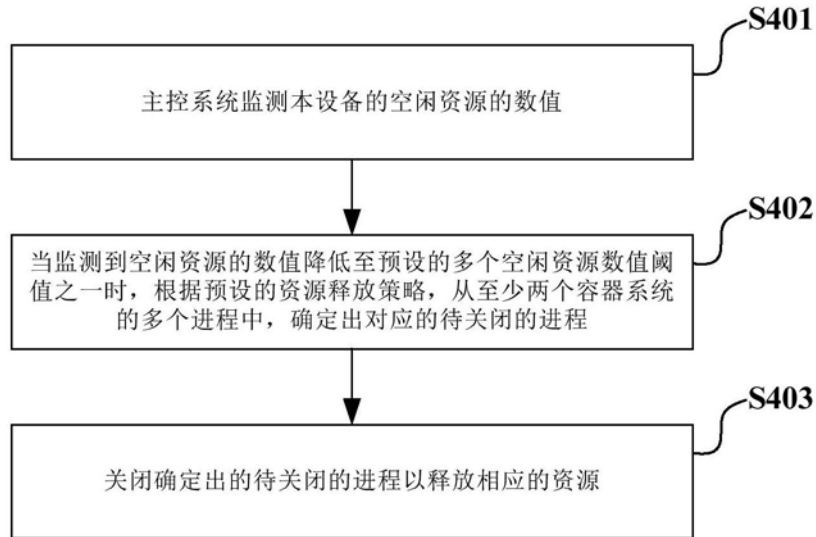


图4

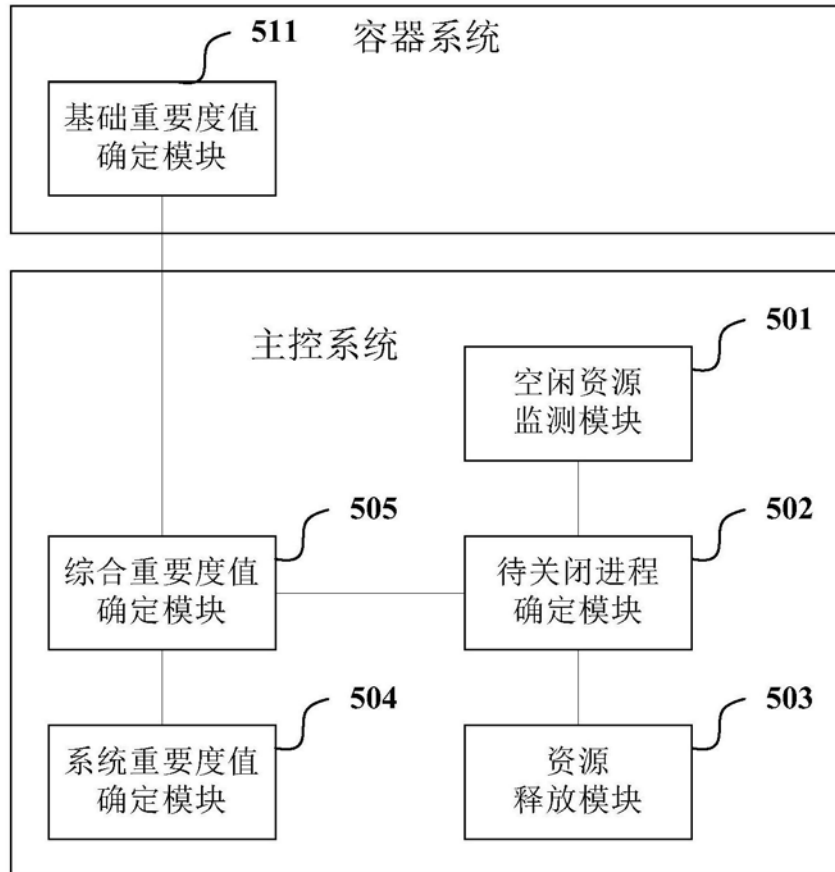


图5

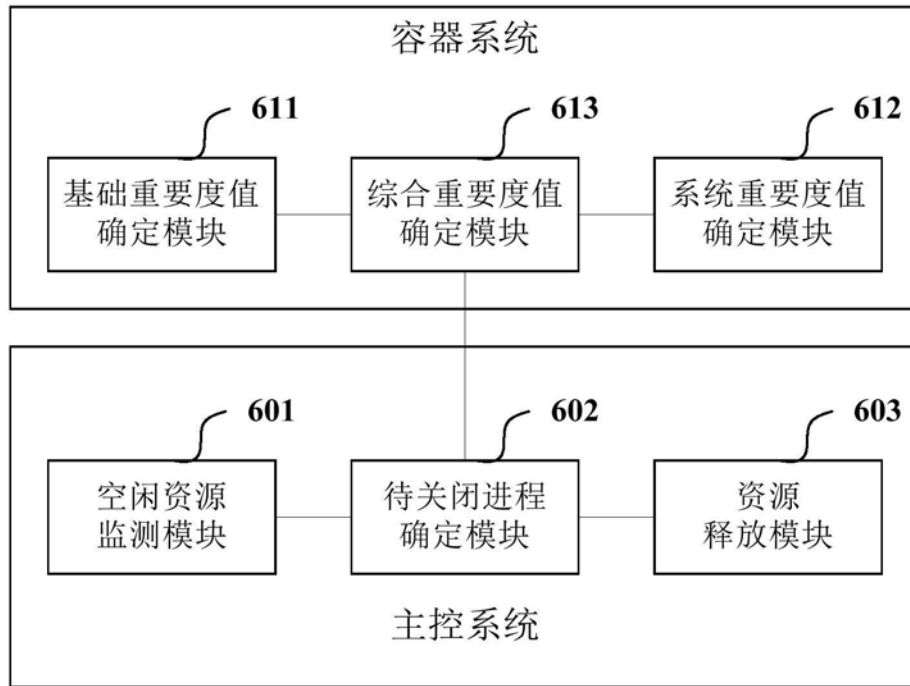


图6