



(12) 发明专利

(10) 授权公告号 CN 116541104 B

(45) 授权公告日 2023.09.29

(21) 申请号 202310811877.8
 (22) 申请日 2023.07.04
 (65) 同一申请的已公布的文献号
 申请公布号 CN 116541104 A
 (43) 申请公布日 2023.08.04
 (73) 专利权人 宁德时代新能源科技股份有限公司
 地址 352100 福建省宁德市蕉城区漳湾镇新港路2号
 专利权人 宁德时代(上海)智能科技有限公司
 (72) 发明人 吴凯 韩志华 王淳
 (74) 专利代理机构 深圳市威世博知识产权代理事务所(普通合伙) 44280
 专利代理师 张庆玲

(51) Int.Cl.
 G06F 9/445 (2018.01)
 G06F 9/4401 (2018.01)
 G06F 8/76 (2018.01)
 (56) 对比文件
 CN 115933578 A, 2023.04.07
 JP 2019096271 A, 2019.06.20
 CN 113641544 A, 2021.11.12
 CN 107608706 A, 2018.01.19
 CN 109062594 A, 2018.12.21
 CN 115828231 A, 2023.03.21
 US 2010186001 A1, 2010.07.22
 韦涌泉 等. 一种基于文件的嵌入式星载软件在轨升级方法. 《单片机与嵌入式系统应用》. 第18卷(第5期), 第32-35页.

审查员 周钟明

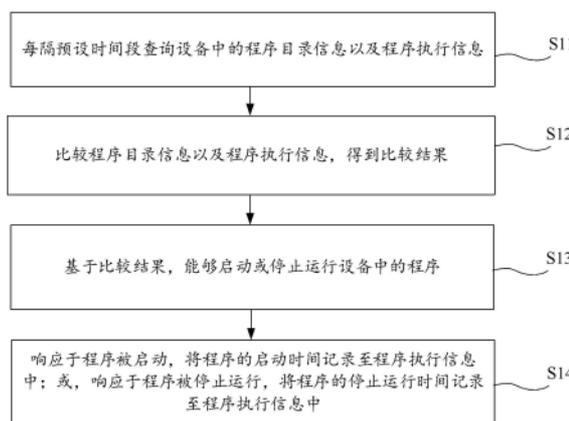
权利要求书2页 说明书9页 附图3页

(54) 发明名称

程序管理方法、控制装置、平台、设备以及存储介质

(57) 摘要

本申请公开了一种程序管理方法、控制装置、平台、设备以及存储介质,方法包括:步骤S11:每隔预设时间段查询设备中的程序目录信息以及程序执行信息,程序目录信息中包括设备中的各程序的相关信息,程序执行信息中包括各程序的启动或停止运行的时间;步骤S12:比较程序目录信息以及程序执行信息,得到比较结果;步骤S13:基于比较结果,能够启动或停止运行设备中的程序。上述方案,能够实现动态启动或停止运行设备中的程序。



1. 一种程序管理方法,其特征在于,包括:

步骤S11:每隔预设时间段查询设备中的程序目录信息以及程序执行信息,所述程序目录信息中包括所述设备中的各程序的相关信息,所述程序执行信息包括各程序的启动或停止运行的时间;

步骤S12:比较所述程序目录信息以及所述程序执行信息,得到比较结果;其中,所述步骤S12包括:基于程序的启动时间或停止运行时间,确定当前处于已启动状态的程序;将所述程序目录信息中的程序与处于已启动状态的程序进行比较,得到比较结果;

步骤S13:基于所述比较结果,能够启动或停止运行所述设备中的程序;其中,所述步骤S13包括:当第一程序为处于已启动状态且不处于所述程序目录信息的程序,能够停止运行第一程序;当第二程序为处于未启动状态且处于所述程序目录信息的程序,能够启动第二程序。

2. 根据权利要求1所述的方法,其特征在于,所述各程序的相关信息包括各程序的配置信息,所述方法还包括:

判断第三程序在所述程序目录信息中的配置信息是否发生变动,所述第三程序为处于已启动状态且处于所述程序目录信息的程序;

响应于所述第三程序的配置信息发生了变动,基于变动后的配置信息能够停止运行第三程序或者重新启动所述第三程序。

3. 根据权利要求2所述的方法,其特征在于,步骤S11还包括:

获取目标程序的当前配置信息;

利用所述当前配置信息更新所述目标程序在所述程序目录信息中的历史配置信息。

4. 根据权利要求2所述的方法,其特征在于,各程序的配置信息中包括所述程序的依赖关系,步骤S13还包括:

基于待启动程序的依赖关系,启动被所述待启动程序依赖的其他程序;

基于待停止运行程序的依赖关系,调整依赖所述待停止运行程序的其他程序的依赖关系。

5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

步骤S14:响应于程序被启动,将程序的启动时间记录至所述程序执行信息中;响应于程序被停止运行,将程序的停止运行时间记录至所述程序执行信息中。

6. 一种控制装置,其特征在于,包括进程管理模块,所述进程管理模块用于执行如权利要求1至5中任一项所述的程序管理方法;其中,所述进程管理模块设置在具备可移植操作系统接口的操作系统上。

7. 一种程序管理平台,其特征在于,包括:

查询单元,用于每隔预设时间段查询设备中的程序目录信息以及程序执行信息,所述程序目录信息中包括所述设备中的各程序的相关信息,所述程序执行信息包括各程序的启动或停止运行的时间;

比较单元,用于比较所述程序目录信息以及所述程序执行信息,得到比较结果;

执行单元,用于基于所述比较结果启动或停止运行所述设备中的程序;

其中,所述比较单元,用于比较所述程序目录信息以及所述程序执行信息,得到比较结果,包括:基于程序的启动时间或停止运行时间,确定当前处于已启动状态的程序,所述程

序处于已启动状态为所述程序处于启动过程中或处于运行过程中的状态;将所述程序目录信息中的程序与处于已启动状态的程序进行比较,得到比较结果;

其中,所述执行单元,用于基于所述比较结果启动或停止运行所述设备中的程序,包括:当第一程序为处于已启动状态且不处于所述程序目录信息的程序,能够停止运行第一程序;当第二程序为处于未启动状态且处于所述程序目录信息的程序,能够启动第二程序。

8.一种计算机设备,其特征在于,包括如权利要求7所述的程序管理平台。

9.一种计算机可读存储介质,其上存储有程序指令,其特征在于,所述程序指令被处理器执行以实现如权利要求1至5中任一项所述的程序管理方法。

程序管理方法、控制装置、平台、设备以及存储介质

技术领域

[0001] 本申请涉及汽车电子领域,特别是涉及一种程序管理方法、控制装置、平台、设备以及存储介质。

背景技术

[0002] 现阶段汽车电子软件设计中,程序的配置文件包含了程序的启动信息,系统能够根据程序的配置文件启动和管理相关的进程。但是目前的方案都是只有在汽车上电时才会读取各程序的配置文件,而在系统运行情况下安装新的程序的话,通常情况下系统无法获取读取到新程序的配置文件,无法启动或管理该新程序,导致无法实现程序的动态启动。

发明内容

[0003] 本申请至少提供一种程序管理方法、控制装置、平台、设备以及存储介质。

[0004] 本申请提供了一种程序管理方法,包括:步骤S11:每隔预设时间段查询设备中的程序目录信息以及程序执行信息,程序目录信息中包括设备中的各程序的相关信息,程序执行信息中包括各程序的启动或停止运行的时间;步骤S12:比较程序目录信息以及程序执行信息,得到比较结果;步骤S13:基于比较结果,能够启动或停止运行设备中的程序。

[0005] 在上述方案中,动态地查询程序目录信息以及程序执行信息,比较程序目标信息以及程序执行信息得到的比较结果能够获知各程序的执行情况,方便根据需要对程序进行管理,例如启动程序或停止运行处于已启动状态的程序,实现程序的动态启动或动态停止运行。

[0006] 在一些实施例中,步骤S12包括:基于各程序的启动时间或停止运行时间,确定当前处于已启动状态的程序;将程序目录信息中的程序与处于已启动状态的程序进行比较,得到比较结果。

[0007] 在上述方案中,通过对各程序进行比较,能够确定哪些程序处于启动状态且不处于程序目录信息,或者处于未启动状态且处于程序目录信息,方便后续基于该比较结果对各程序进行针对性的处理。

[0008] 在一些实施例中,步骤S13包括:当第一程序为处于已启动状态且不处于程序目录信息的程序,能够停止运行第一程序;或,当第二程序为处于未启动状态且处于程序目录信息的程序,能够启动第二程序。

[0009] 在上述方案中,程序执行信息中包括各程序的启动或停止运行的时间,通过各程序的启动时间或停止运行时间能够确定各程序处于已启动状态还是处于停止运行状态,从而根据比较的结果停止运行或启动对应的程序,以此实现动态启动或停止运行对应的程序。

[0010] 在一些实施例中,各程序的相关信息包括各程序的配置信息,方法还包括:判断第三程序在程序目录信息中的配置信息是否发生变动,第三程序为处于已启动状态且处于程序目录信息的程序;响应于第三程序的配置信息发生了变动,基于变动后的配置信息能够

停止运行第三程度程序或者重新启动第三程序。

[0011] 在上述方案中,通过判断已经启动的程序的配置信息是否发生变动,若配置信息发生变动则重新启动对应的程序,方便运行更新后的程序。

[0012] 在一些实施例中,方法包括:获取目标程序的当前配置信息;利用当前配置信息更新目标程序在程序目录信息中的历史配置信息。

[0013] 在上述方案中,通过获取目标程序的当前配置信息,然后能够利用当前配置信息更新目标程序在程序目录信息中的历史配置信息,方便后续确定程序目标信息中程序的配置信息是否发生变动。

[0014] 在一些实施例中,各程序的配置信息中包括程序的依赖关系,步骤S13还包括:基于待启动程序的依赖关系,启动被待启动程序依赖的其他程序;或,基于待停止运行程序的依赖关系,调整依赖待停止运行程序的其他程序的依赖关系。

[0015] 在上述方案中,通过确定待启动程序的依赖关系,启动需要依赖的其他程序,保障待启动程序的正常启动或正常运行,通过调整依赖待停止运行程序的其他程序的依赖关系,以防待停止运行程序停止运行之后,其他程序的运行错误。

[0016] 在一些实施例中,方法还包括:步骤S14:响应于程序被启动,将程序的启动时间记录至程序执行信息中;或者,响应于程序被停止运行,将程序的停止运行时间记录至程序执行信息中。

[0017] 在上述方案中,程序被启动时将程序的启动时间记录至程序执行信息中,或者程序被停止运行时将程序的停止运行时间记录在程序执行信息中,方便后续判断程序是处于已启动状态还是停止运行状态。

[0018] 本申请提供一种控制装置,包括进程管理模块,进程管理模块用于执行上述任意程序管理方法,其中,进程管理模块设置在具备可移植操作系统接口的操作系统上。

[0019] 在上述方案中,通过设置可移植操作系统接口,保证不同操作系统之间的兼容性。

[0020] 本申请提供了一种平台,尤其是自适应平台,包括:查询单元、比较单元以及执行单元;查询单元,用于每隔预设时间段查询,尤其是自动地查询设备中的程序目录信息以及程序执行信息,程序目录信息中包括设备中的各程序的相关信息,程序执行信息中包括各程序的启动或停止运行的时间;比较单元,用于比较程序目录信息以及程序执行信息,得到比较结果;执行单元,用于基于比较结果,能够启动或停止运行设备中的程序。

[0021] 本申请提供了一种设备,包括上述平台。

[0022] 本申请提供了一种计算机可读存储介质,其上存储有程序指令,程序指令被处理器执行时实现上述程序管理方法。

[0023] 在上述方案中,动态地查询程序目录信息以及程序执行信息,比较程序目标信息以及程序执行信息得到的比较结果能够获知各程序的执行情况,方便根据需要对程序进行管理,例如启动程序或停止运行处于已启动状态的程序,实现程序的动态启动或动态停止运行。

[0024] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,而非限制本申请。

附图说明

[0025] 此处的附图被并入说明书中并构成本说明书的一部分,这些附图示出了符合本申请的实施例,并与说明书一起用于说明本申请的技术方案。

[0026] 图1是一些实施例提供的程序管理方法的流程示意图;

[0027] 图2是一些实施例提供的程序目录文件的结构示意图;

[0028] 图3是一些实施例提供的程序管理方法的另一流程示意图;

[0029] 图4是一些实施例提供的控制装置的结构示意图;

[0030] 图5是一些实施例提供的进程管理模块的部署示意图;

[0031] 图6是一些实施例提供的平台的结构示意图;

[0032] 图7是一些实施例提供的设备的结构示意图;

[0033] 图8是一些实施例提供的计算机可读存储介质的结构示意图。

具体实施方式

[0034] 下面结合说明书附图,对本申请实施例的方案进行详细说明。

[0035] 以下描述中,为了说明而不是为了限定,提出了诸如特定子系统结构、接口、技术之类的具体细节,以便透彻理解本申请。

[0036] 本文中术语“和/或”,仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,本文中字符“/”,一般表示前后关联对象是一种“或”的关系。此外,本文中的“多”表示两个或者多于两个。另外,本文中术语“至少一种”表示多种中的任意一种或多种中的至少两种的任意组合,例如,包括A、B、C中的至少一种,可以表示包括从A、B和C构成的集合中选择的任意一个或多个元素。

[0037] 现阶段汽车电子软件设计中,自适应平台(AP)的进程管理模块(EM模块)可以做到根据程序的配置文件启动和管理相关的进程,但其启动进程是根据配置文件静态的启动,如果系统中安装新的应用程序或者利用空中下载技术(Over-the-Air Technology,OTA)增加新的应用程序,则没有办法直接进行动态启动和管理。故,本方案提供了一种能够动态启动或关闭程序的方式。

[0038] 请参阅图1,本申请提供的程序管理方法可以包括以下步骤S11至步骤S13的内容。步骤S11:每隔预设时间段查询设备中的程序目录信息以及程序执行信息。其中,程序目录信息中包括设备中的各程序的相关信息,程序执行信息中包括各程序的启动或停止运行的时间。步骤S12:比较程序目录信息以及程序执行信息,得到比较结果。步骤S13:基于比较结果,能够启动或停止运行设备中的程序。

[0039] 设备可以是车辆、轮船、飞机等任意需要运行程序的交通设备或者工业设备等任意需要运行程序的设备。本实施例以设备为车辆为例。程序可以是应用程序,也可以是系统程序。预设时间段的长度的设置方式包括但不限于用户自定义或根据设备内所包含的至少部分程序的大小确定。示例性地,若设备中最小程序的大小为A,下载该程序的时间为B,则预设时间段可以是B。程序目录信息中包括设备中各程序的相关信息可以认为是关于设备所包含程序的标识信息和/或配置信息等等,示例性地,程序目录信息可以包含在程序目录文件中,该文件下又可包括各程序的相关配置文件。如图2所示,程序目录文件的名称为

PRO_ROOT,下面包括多种可执行配置文件,例如包括进程管理模块EM的相关文件、如状态管理模块SM的相关文件以及各程序的相关文件,各程序可以是APP1、APP2以及APP3等,相关文件可以是APP1、APP2以及APP3的可执行文件(EXE文件)、配置文件(CONFIG文件)以及日志文件(LOG文件)等。一些应用场景中,程序的相关文件可以是该程序下不同功能模块(例如A功能模块、B功能模块)的相关文件。程序执行信息可以是程序执行记录,例如可以是程序执行日志。日志可以记录各程序状态的变动时间。各程序的启动时间一般可以包括处于启动状态的程序开始该启动状态的时间,程序的停止运行时间可以包括处于停止运行的程序停止运行时的时间。通过比较程序目录信息和程序执行信息能够确定程序目录信息中包含的程序当前处于已启动状态还是停止运行状态。比较结果可以是程序目录信息中所有的程序都已经启动或部分程序处于未已启动状态,或者已经启动的程序不在程序目录中。基于比较结果,能够启动或停止运行设备中的程序的方式可以是基于比较结果启动处于未启动状态的程序或停止运行处于已启动状态的程序。

[0040] 在上述方案中,动态地查询程序目录信息以及程序执行信息,比较程序目标信息以及程序执行信息得到的比较结果能够获知各程序的执行情况,方便根据需要对程序进行管理,例如启动程序或停止运行处于已启动状态的程序,实现程序的动态启动或动态停止运行。

[0041] 在一些实施例中,上述步骤S12可以包括:基于程序的执行记录,确定当前处于已启动状态的程序。尤其是最新的操作记录,例如程序的最新记录是包括程序启动时间或停止运行时间,那么就表示程序处于运行状态或停止状态,确定当前处于已启动状态的程序。然后,将程序目录信息中的程序与处于已启动状态的程序进行比较,得到比较结果。

[0042] 上述基于各程序的启动时间或停止运行时间,确定当前处于已启动状态的程序的方式可以是基于该程序在距离当前时刻最近的记录确定。示例性地,若该程序在距离当前时刻最近一次记录的时间为启动时间,则可以确定该程序已经启动,同理,若该程序在距离当前时刻最近一次记录的时间为停止运行时间,则可以确定该程序处于未启动状态。程序处于已启动状态具体可以理解为程序处于启动过程中或处于运行过程中的状态。

[0043] 在上述方案中,通过对各程序进行比较,能够确定哪些程序处于启动状态且不处于程序目录信息,或者处于未启动状态且处于程序目录信息,方便后续基于该比较结果对各程序进行针对性的处理。

[0044] 在一些实施例中,上述步骤S13可以包括:当第一程序为处于已启动状态且不处于程序目录信息的程序,能够停止运行第一程序;或,当第二程序为处于未启动状态且处于程序目录信息的程序,能够启动第二程序。

[0045] 也就是,处于已启动状态且不处于程序目录信息中的程序都可以称之为第一程序。若程序已经启动,但是该程序不处于程序目录信息中,则表明该程序在启动之后很可能已经被云端或用户从设备中删除,则可以停止运行该程序。处于未启动状态且处于程序目录信息的程序都可以称之为第二程序。处于未启动状态且处于程序目标信息的程序可能是新增到设备中的程序,则可以启动该程序。

[0046] 在上述方案中,程序执行信息中包括各程序的启动或停止运行的时间,通过各程序的启动时间或停止运行时间等历史信息能够确定各程序处于已启动状态还是处于停止运行状态,从而根据比较的结果能够停止运行或启动对应的程序,以此实现动态启动或停

止运行对应的程序。

[0047] 在一些实施例中,各程序的相关信息包括各程序的配置信息。方法还包括:判断第三程序在程序目录信息中的配置信息是否发生变动,第三程序为处于已启动状态且处于程序目录信息的程序;响应于第三程序的配置信息发生了变动,基于变动后的配置信息能够停止运行或重新启动第三程序。

[0048] 配置信息一般用于指示程序启动或运行所需的配置参数等内容。程序在程序目录信息中的配置信息发生变动之后,很可能意味着该程序被更新了,此时可以重新启动该程序以便使用最新的配置信息启动运行该程序。判断第三程序在程序目录信息中的配置信息是否发生变动的方式可以是判断程序的当前配置信息与上一次读取的历史配置信息之间是否存在区别,若存在区别则说明程序的配置信息发生变动,或者还可以是根据程序当前启动或运行使用的配置参数与程序目录信息下该程序的配置信息是否存在区别,若存在区别则说明程序的配置信息发生变动。

[0049] 在上述方案中,通过判断已经启动的程序的配置信息是否发生变动,若配置信息发生变动则停止程序运行,或者,重新启动对应的程序,方便运行更新后的程序。

[0050] 在一些实施例中,上述步骤S11还包括:获取目标程序的当前配置信息。利用当前配置信息更新目标程序在程序目录信息中的历史配置信息。

[0051] 目标程序可以从外部接口接收到的下载安装包或更新安装包对应的程序。也就是目标程序可以是新增到设备中的程序也可以是设备中需要更新升级的程序。获取目标程序的当前配置信息的方式可以从程序对应的下载安装包解析得到。利用当前配置信息更新目标程序在程序目录信息中的历史配置信息之前,先判断程序目录信息中是否包括该程序,若包括则可以更新历史配置信息,若不包括则可以直接将获取到的目标程序的当前配置信息作为程序目录信息中该程序的配置信息,示例性地,在程序目录文件下新建关于目标程序的文件,然后将目标程序的配置信息放入目标程序对应的文件下。如图3所示,可以通过OTA的方式或者外部输入接口(包括有线方式,例如串口、网口,以及无线方式,例如蓝牙、WIFI)的方式导入新的程序(例如导入应用程序APP)。

[0052] 在上述方案中,通过获取目标程序的当前配置信息,然后能够利用当前配置信息更新目标程序在程序目录信息中的历史配置信息,方便后续确定程序目标信息中程序的配置信息是否发生变动。

[0053] 在一些实施例中,各程序的配置信息中包括程序的依赖关系。上述步骤S13还包括:基于待启动程序的依赖关系,启动被待启动程序依赖的其他程序。或,基于待停止运行程序的依赖关系,调整依赖待停止运行程序的其他程序的依赖关系。

[0054] 待启动程序可以是需要启动的程序,如上述第二程序或第三程序,待停止运行程序可以是需要停止运行的程序,如上述第一程序。调整依赖关系的方式包括但不限于直接删除该依赖关系或者确定替代程序并建立替代程序与依赖待停止运行程序之间的依赖关系。一些应用场景中,部分程序之间的启动或运行存在依赖关系,例如A程序的运行需要应用到B程序的某功能,也就是A程序依赖于B程序,B程序被A程序依赖,此时,若A程序为待启动程序,则根据该依赖关系,可以启动被待启动程序依赖的B程序。示例性地,可以在启动待启动程序之前,启动被待启动程序依赖的其他程序。一些应用场景中,若待停止运行程序停止运行,很可能导致需要依赖待停止程序的其他程序的运行或启动产生问题,则可以更新

需要依赖待停止运行程序的其他程序的依赖关系。示例性地,从该待停止运行程序以外的其他程序中确定替代程序以调整其他程序的依赖关系,或者直接将需要依赖该待停止运行程序的其他程序与该待停止运行程序的依赖关系删除。

[0055] 在上述方案中,通过确定待启动程序的依赖关系,启动需要依赖的其他程序,保障待启动程序的正常启动或正常运行;通过调整依赖待停止运行程序的其他程序的依赖关系,以防待停止运行程序停止运行之后,导致其他程序的运行错误。

[0056] 在一些实施例中,方法还包括:步骤S14:响应于程序被启动,将程序的启动时间记录至程序执行信息中;或,响应于程序被停止运行,将程序的停止运行时间记录至程序执行信息中。

[0057] 具体可以通过日志的方式记录各程序的启动时间和停止运行时间。各程序的执行信息可以记录在一个日志里面,也可以分别为各程序建立对应的日志用于分别记录对应程序的执行信息。

[0058] 在上述方案中,程序被启动时将程序的启动时间记录至程序执行信息中,或者程序被停止运行时将程序的停止运行时间记录在程序执行信息中,方便后续判断程序是处于已启动状态还是停止运行状态。

[0059] 请参阅图4,本申请提供的控制装置30中包括进程管理模块31。进程管理模块31用于执行上述任意的程序管理方法。其中,进程管理模块31设置在具备可移植操作系统接口的操作系统上。

[0060] 其中,控制装置30可以是微处理器。示例性地,控制装置30可以应用于自适应平台等能够实现本实施例提供的程序管理方法的平台。相比AUTOSAR经典平台,自适应平台运行具有多核的强大微处理器,例如微处理器可以具有1 GHz以上主频,并且可以访问更多的内存(64MB到2GB)。自适应平台通过软、硬件的结合提供了开发高灵活性、高效率应用的独特功能,使得开发者能够专注于应用的具体差异化。进程管理模块31(Execution Management,简称EM)可以认为是一个功能集群,负责系统执行管理的各个方面,包括平台初始化和启动/关闭应用程序。也就是进程管理模块31执行上述步骤S11至步骤S13的内容。可移植操作系统接口(Portable Operating System Interface of UNIX,缩写为POSIX)。操作系统包括但不限于OS操作系统或Linux操作系统等。一些应用场景中,进程管理模块31(EM)在一个节点上(MCU、独立运行的OS上、SOC上的独立容器内)仅运行一个实例,换言之,在一个节点上只使用一个进程管理模块31来管理各程序。进程管理模块31(EM)部署在POSIX操作系统之上,并基于操作系统,实现进程调度功能。如图5所示,进程管理模块31(EM)部署在通讯中间件之上,并基于通讯中间件,实现进程间服务发现、信息交互等功能。进程管理模块31(EM)和其余Function API一起(如状态管理模块SM、诊断模块DIAG、数据库模块DB等),实现整体的系统基础服务功能。

[0061] 在上述方案中,通过设置可移植操作系统接口,保证不同操作系统之间的兼容性。

[0062] 在一些实施例中,为实现动态发现并启动或者停止进入系统新的应用程序的方法,还提供以下实施例。

[0063] 如图3所示,可以使用OTA或者其他外部输入的方式将程序相关文件载入,并将新的程序文件新增至程序目录文件中。管理模块通过动态扫描的方式能够发现程序是否有更新或改变,能够实现动态启动或者停止相应的应用程序,达到动态管理程序的效果,而不用

像现有技术中采用的每次更新后都需要重新上电才能启用。

[0064] 进程管理模块是操作系统启动之后,被执行的第一个守护进程。它负责根据配置信息,以进程的方式启动执行和配置每个可执行文件(APP)。进程管理模块的主要功能,是管理和控制服务/软件(APP)的启动、关闭,并动态扫描一个固定文件结构目录下的文件,固定文件结构目录可以认为是上述程序目录文件,程序目录文件携带的信息可以包含程序目录信息,固定文件结构目录下的文件可以包括各程序对应的相关文件,各程序的相关文件可以包含各程序的相关信息。当有扫描到新的文件时,查询相应的依赖、配置并能够根据需要启动新加入的APP,若有被删除的应用程序,则负责更新依赖并停止相应的应用程序。其中,动态扫描可以用shell脚本或者代码实现,通过读取固定目录下文件是否有增加新的文件或者删除文件来确定变更。

[0065] 其中,进程管理可执行的配置文件和应用程序的配置文件以JSON文件格式提供。如图2所示,程序目录文件下还包括进程管理模块的配置文件、状态管理模块的配置文件。并在APP/EM集成发布期间同时发布。配置文件内的信息包括:可执行文件的启动条件、版本、启动模式、资源需求、调度优先级、可提供服务接口(非必需)和/或依赖性服务接口(非必需)等。

[0066] 进程管理模块EM在运行期间,动态维护一个JSON文件格式的记录文件,该记录文件记录了程序执行信息,也就是执行记录文件内存储关于本节点范围内的进程启动或停止的日志,用于与实时扫描目录下的文件比较,来判断新的应用程序的导入。动态维护的方式可以是动态扫描。

[0067] 上述方案提供的进程管理模块能够应用于汽车嵌入式SOA架构(Service Oriented Architecture,面向服务的架构)。

[0068] 请参阅图6,如图6所示,本实施例提供的平台40可以包括:查询单元41、比较单元42以及执行单元43。平台40具体可以是自适应平台或其他能够实现本申请实施例提供的程序管理方法的平台,例如AUTOSAR经典平台等。查询单元41,用于每隔预设时间段查询设备中的程序目录信息以及程序执行信息,程序目录信息中包括设备中的各程序的相关信息,程序执行信息中包括各程序的启动或停止运行的时间;比较单元42,用于比较程序目录信息以及程序执行信息,得到比较结果;执行单元43,用于基于比较结果,能够启动或停止运行设备中的程序。

[0069] 可选地,查询单元、比较单元以及执行单元可部署在上述进程管理模块内。换言之,由自适应平台中的进程管理模块执行每隔预设时间段查询,尤其是自动地查询设备中的程序目录信息以及程序执行信息的步骤、比较程序目录信息以及程序执行信息,得到比较结果的步骤以及基于比较结果,能够启动或停止运行设备中的程序的步骤。

[0070] 在上述方案中,动态地查询程序目录信息以及程序执行信息,比较程序目标信息以及程序执行信息,根据得到的比较结果能够得到各程序的执行情况,方便根据各程序的执行情况启动其他的程序或停止运行处于已启动状态的程序,实现程序的动态启动或动态停止运行。

[0071] 在一些实施例中,比较单元42比较程序目录信息以及程序执行信息,得到比较结果,包括:基于各程序的启动时间或停止运行时间,确定当前处于已启动状态的程序;将程序目录信息中的程序与处于已启动状态的程序进行比较,得到比较结果。

[0072] 在上述方案中,通过对各程序进行比较,能够确定哪些程序处于启动状态且不处于程序目录信息,或者处于未启动状态且处于程序目录信息,方便后续基于该比较结果对各程序进行针对性的处理。

[0073] 在一些实施例中,执行单元43基于比较结果能够选择启动或停止运行设备中的程序,包括:当第一程序为处于已启动状态且不处于程序目录信息的程序,能够停止运行第一程序;或,当第二程序为处于未启动状态且处于程序目录信息的程序,能够启动第二程序。

[0074] 在上述方案中,程序执行信息中包括各程序的启动或停止运行的时间,通过各程序的启动时间或停止运行时间能够确定各程序处于已启动状态还是处于停止运行状态,从而根据比较的结果停止运行或启动对应的程序,以此实现动态启动或停止运行对应的程序。

[0075] 在一些实施例中,各程序的相关信息包括各程序的配置信息,比较单元42还用于:判断第三程序在程序目录信息中的配置信息是否发生变动,第三程序为处于已启动状态且处于程序目录信息的程序;响应于第三程序的配置信息发生了变动,基于变动后的配置信息停止第三程序或者重新启动第三程序。

[0076] 在上述方案中,通过判断已经启动的程序的配置信息是否发生变动,若配置信息发生变动则可以重新启动对应的程序,方便运行更新后的程序;或者,若配置信息发生变动导致当前程序需要停止时,则可以停止对应的程序。

[0077] 在一些实施例中,查询单元41还用于:获取目标程序的当前配置信息;利用当前配置信息更新目标程序在程序目录信息中的历史配置信息。

[0078] 在上述方案中,通过获取目标程序的当前配置信息,然后能够利用当前配置信息更新目标程序在程序目录信息中的历史配置信息,方便后续确定程序目标信息中程序的配置信息是否发生变动。

[0079] 在一些实施例中,各程序的配置信息中包括程序的依赖关系,执行单元43还用于:基于待启动程序的依赖关系,启动被待启动程序依赖的其他程序;或,基于待停止运行程序的依赖关系,调整依赖待停止运行程序的其他程序的依赖关系。

[0080] 在上述方案中,通过确定待启动程序的依赖关系,启动需要依赖的其他程序,保障待启动程序的正常启动或正常运行,通过调整依赖待停止运行程序的其他程序的依赖关系,以防待停止运行程序停止运行之后,导致其他程序的运行错误。

[0081] 在一些实施例中,平台40还包括状态管理单元(图中未示出),状态管理单元可以部署在状态管理模块中,状态管理单元用于:响应于程序被启动,将程序的启动时间记录至程序执行信息中;或,响应于程序被停止运行,将程序的停止运行时间记录至程序执行信息中。

[0082] 在上述方案中,程序被启动时将程序的启动时间记录至程序执行信息中,或者程序被停止运行时将程序的停止运行时间记录在程序执行信息中,方便后续判断程序是处于已启动状态还是停止运行状态。

[0083] 请参阅图7,本实施例提供的设备50包括上述平台40。

[0084] 本实施例提供的设备50可以是车辆、轮船、飞机等任意需要运行程序的交通设备或者工业设备等任意需要运行程序的设备。

[0085] 在上述方案中,通过动态地查询程序目录信息以及程序执行信息,比较程序目标

信息以及程序执行信息得到的比较结果能够获知各程序的执行情况,方便根据各程序的执行情况启动其他的程序或停止运行处于已启动状态的程序,实现程序的动态启动或动态停止运行。

[0086] 请参阅图8,本实施例提供的计算机可读存储介质60存储有能够被处理器运行的程序指令61,程序指令61用于实现上述任一程序管理方法实施例中的步骤。

[0087] 上述方案,动态地查询程序目录信息以及程序执行信息,比较程序目标信息以及程序执行信息得到的比较结果能够获知各程序的执行情况,方便根据需要对程序进行管理,例如启动程序或停止运行处于已启动状态的程序,实现程序的动态启动或动态停止运行。上文对各个实施例的描述倾向于强调各个实施例之间的不同之处,其相同或相似之处可以互相参考,为了简洁,本文不再赘述。

[0088] 在本申请所提供的几个实施例中,应该理解到,所揭露的方法和装置,可以通过其它的方式实现。例如,以上所描述的装置实施方式仅仅是示意性的,例如,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,例如单元或组件可以结合或者可以集成到另一个子系统,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信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性、机械或其它的形式。

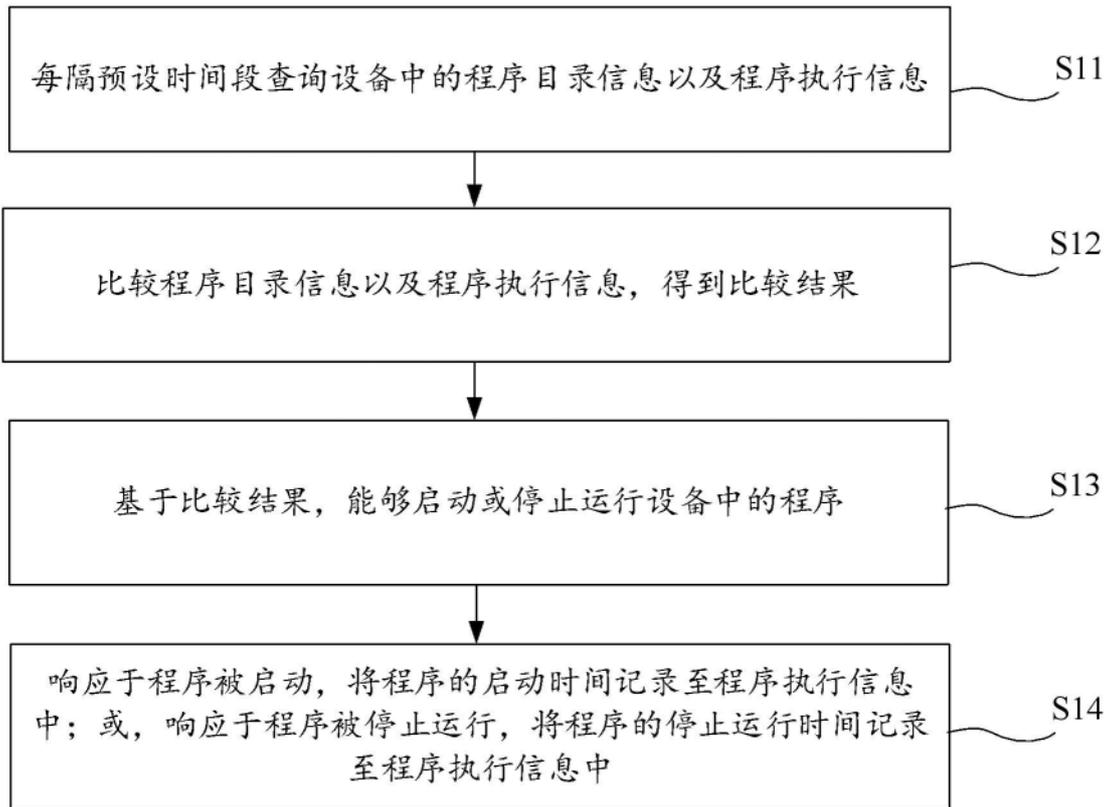


图1

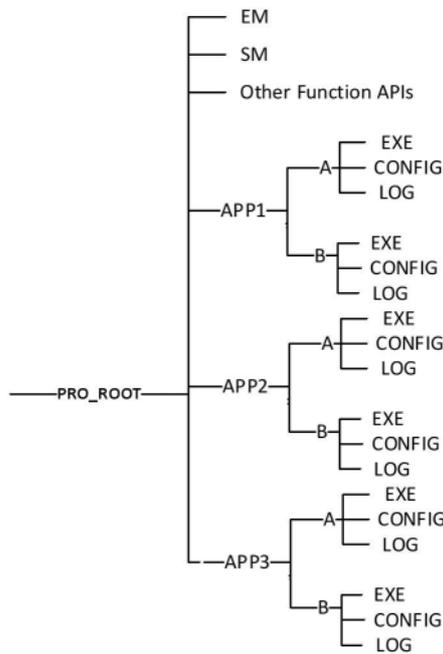


图2

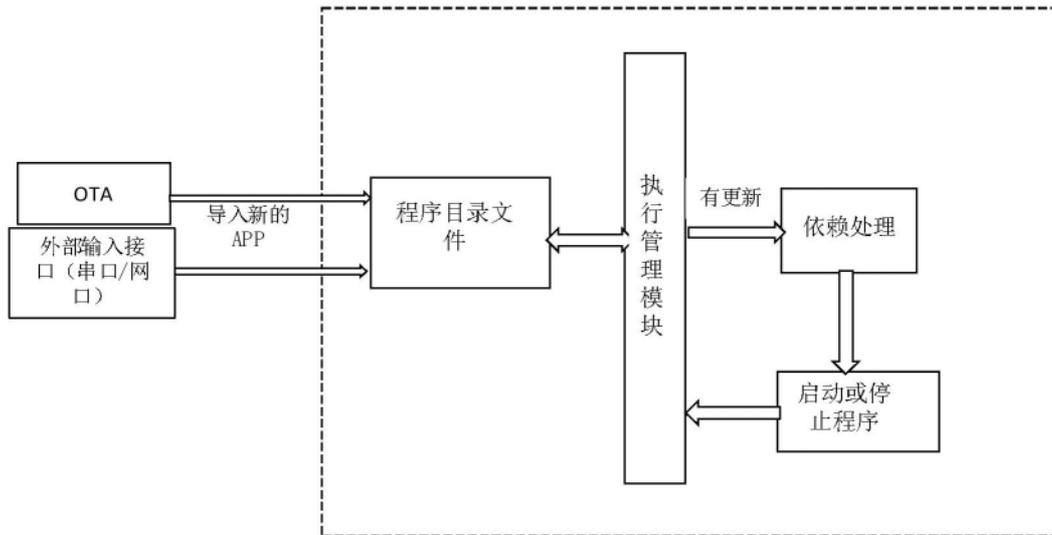


图3

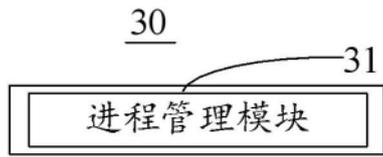


图4

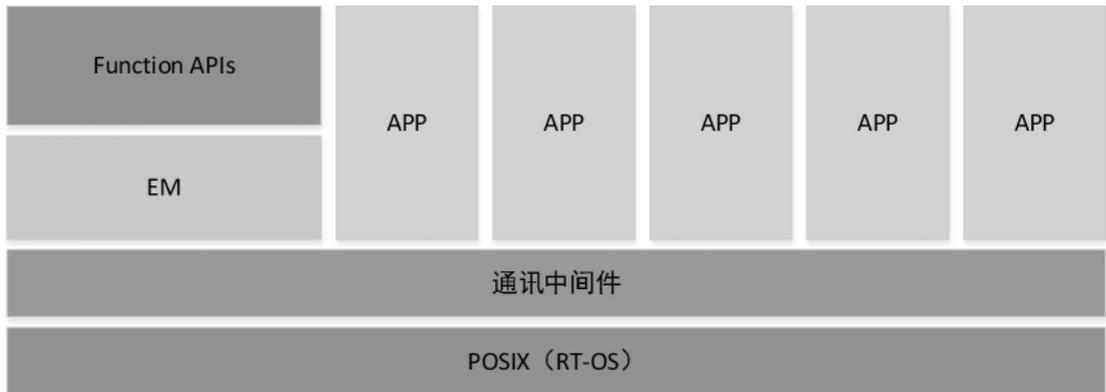


图5

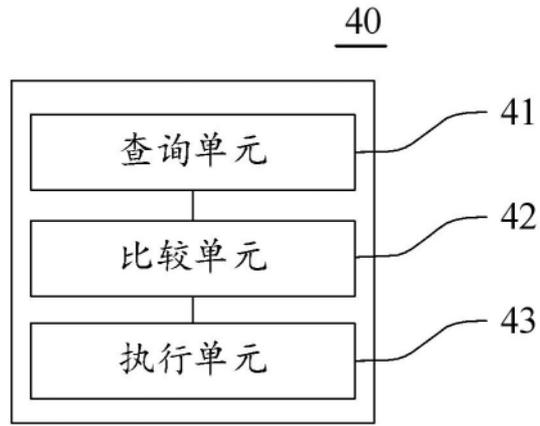


图6

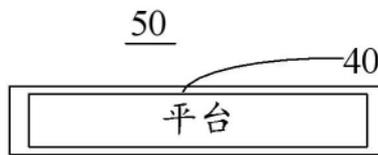


图7

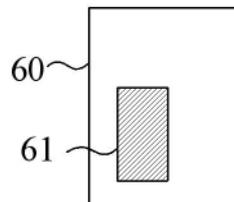


图8