



(12) 发明专利

(10) 授权公告号 CN 113254240 B

(45) 授权公告日 2021.10.15

(21) 申请号 202110683953.2

(22) 申请日 2021.06.21

(65) 同一申请的已公布的文献号  
申请公布号 CN 113254240 A

(43) 申请公布日 2021.08.13

(73) 专利权人 苏州浪潮智能科技有限公司  
地址 215100 江苏省苏州市吴中区吴中经  
济开发区郭巷街道官浦路1号9幢

(72) 发明人 张彭

(74) 专利代理机构 北京连和连知识产权代理有  
限公司 11278

代理人 刘小峰 张涛

(51) Int. Cl.  
G06F 9/54 (2006.01)

(56) 对比文件

CN 109669724 A, 2019.04.23

CN 108196794 A, 2018.06.22

CN 110008708 A, 2019.07.12

审查员 鱼冰

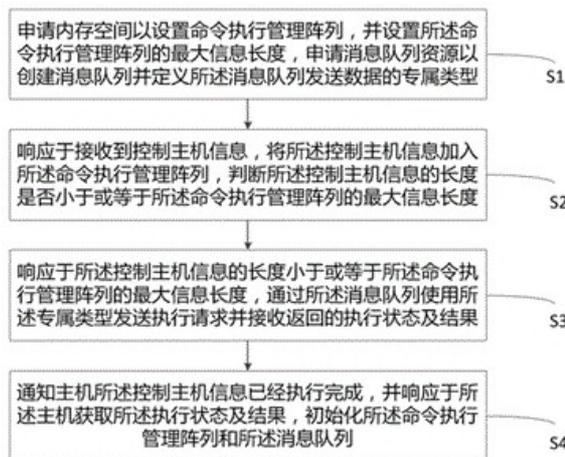
权利要求书2页 说明书7页 附图2页

(54) 发明名称

一种管理控制设备的方法、系统、设备及介  
质

(57) 摘要

本发明公开了一种管理控制设备的方法、系  
统、设备和存储介质,方法包括:申请内存空间以  
设置命令执行管理阵列,并设置命令执行管理阵  
列的最大信息长度,申请消息队列资源以创建消  
息队列并定义消息队列发送数据的专属类型;响  
应于接收到控制主机信息,将控制主机信息加入  
命令执行管理阵列,并判断控制主机信息的长度  
是否小于或等于命令执行管理阵列的最大信息  
长度;响应于控制主机信息的长度小于或等于命  
令执行管理阵列的最大信息长度,通过消息队列  
使用专属类型发送执行请求并接收返回的执行  
状态及结果;以及通知主机控制主机信息已经执  
行完成,并响应于主机获取执行状态及结果,初  
始化命令执行管理阵列和消息队列。



1. 一种管理控制设备的方法,其特征在于,包括以下步骤:

申请内存空间以设置命令执行管理阵列,并设置所述命令执行管理阵列的最大信息长度,申请消息队列资源以创建消息队列并定义所述消息队列发送数据的专属类型;

响应于接收到控制主机信息,将所述控制主机信息加入所述命令执行管理阵列,并判断所述控制主机信息的长度是否小于或等于所述命令执行管理阵列的最大信息长度;

响应于所述控制主机信息的长度小于或等于所述命令执行管理阵列的最大信息长度,通过所述消息队列使用所述专属类型发送执行请求,并接收返回的执行状态及结果;以及

通知主机所述控制主机信息已经执行完成,并响应于所述主机获取所述执行状态及结果,初始化所述命令执行管理阵列和所述消息队列。

2. 根据权利要求1所述的方法,其特征在于,方法还包括:

响应于所述命令执行管理阵列设置完成,启用新线程监听所述消息队列以检测是否存在执行请求。

3. 根据权利要求1所述的方法,其特征在于,方法还包括:

响应于接收到所述执行请求,在当前进程下调用fork函数创建子进程,在所述子进程中调用execve函数执行所述执行请求。

4. 根据权利要求3所述的方法,其特征在于,方法还包括:

在所述当前进程和所述子进程间设置无管道,并进行重定向以接收所述execve函数执行的状态及结果。

5. 根据权利要求1所述的方法,其特征在于,方法还包括:

动态申请内存空间保存所述执行的状态及结果,并将所述内存空间的地址写入所述命令执行管理阵列。

6. 根据权利要求1所述的方法,其特征在于,方法还包括:

根据当前设备型号或场景判断是否需要创建命令执行管理阵列和/或消息队列;以及响应于需要创建命令执行管理阵列和/或消息队列,发送使能信号并创建命令处理线程。

7. 一种管理控制设备的系统,其特征在于,包括:

创建模块,配置用于申请内存空间以设置命令执行管理阵列,并设置所述命令执行管理阵列的最大信息长度,申请消息队列资源以创建消息队列并定义所述消息队列发送数据的专属类型;

判断模块,配置用于响应于接收到控制主机信息,将所述控制主机信息加入所述命令执行管理阵列,并判断所述控制主机信息的长度是否小于或等于所述命令执行管理阵列的最大信息长度;

消息模块,配置用于响应于所述控制主机信息的长度小于或等于所述命令执行管理阵列的最大信息长度,通过所述消息队列使用所述专属类型发送执行请求,并接收返回的执行状态及结果;以及

初始化模块,配置用于通知主机所述控制主机信息已经执行完成,并响应于所述主机获取所述执行状态及结果,初始化所述命令执行管理阵列和所述消息队列。

8. 一种计算机设备,其特征在于,包括:

至少一个处理器;以及

存储器,所述存储器存储有可在所述处理器上运行的计算机指令,所述指令由所述处理器执行时实现权利要求1-6任意一项所述方法的步骤。

9.一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1-6任意一项所述方法的步骤。

## 一种管理控制设备的方法、系统、设备及介质

### 技术领域

[0001] 本发明涉及Linux系统领域,更具体地,特别是指一种管理控制设备的方法、系统、计算机设备及可读介质。

### 背景技术

[0002] Linux操作系统,特别是其嵌入式系统,由于其源代码的可开发性、成本低廉、系统稳定可靠以及可裁剪等特点,如今成为主流的嵌入式操作系统之一,已被广泛应用于工业制造、过程控制、通讯、仪器、仪表、汽车、船舶、航空、航天、军事装备、消费类产品等众多领域。

[0003] Linux操作系统的应用离不开处理器硬件平台,大量、复杂的需求带来了各种各样处理器外围芯片设备的广泛应用,采用现有的Linux操作系统的设备驱动开发的逻辑结构,同一总线如果挂接多个设备,并且多个设备同时访问的话,总线互斥访问很难做到,并且现有对设备的控制主要通过控制数据流,而控制数据流又往往会对业务数据流产生影响,使得业务处理的效率降低。

### 发明内容

[0004] 有鉴于此,本发明实施例的目的在于提出一种管理控制设备的方法、系统、计算机设备及计算机可读存储介质,本发明采用单线程对命令执行管理阵列和消息队列逐层进行处理,减少了控制数据流对业务数据流的影响,然后通过线程间通信机制实现各层间控制数据的交流,以实现对于Linux系统设备的管理,在避免对业务产生影响的同时,实现对设备的管理和操控。

[0005] 基于上述目的,本发明实施例的一方面提供了一种管理控制设备的方法,包括如下步骤:申请内存空间以设置命令执行管理阵列,并设置所述命令执行管理阵列的最大信息长度,申请消息队列资源以创建消息队列并定义所述消息队列发送数据的专属类型;响应于接收到控制主机信息,将所述控制主机信息加入所述命令执行管理阵列,并判断所述控制主机信息的长度是否小于或等于所述命令执行管理阵列的最大信息长度;响应于所述控制主机信息的长度小于或等于所述命令执行管理阵列的最大信息长度,通过所述消息队列使用所述专属类型发送执行请求并接收返回的执行状态及结果;以及通知主机所述控制主机信息已经执行完成,并响应于所述主机获取所述执行状态及结果,初始化所述命令执行管理阵列和所述消息队列。

[0006] 在一些实施方式中,方法还包括:响应于所述命令执行管理阵列设置完成,启用新线程监听所述消息队列以检测是否存在执行请求。

[0007] 在一些实施方式中,方法还包括:响应于接收到所述执行请求,在当前进程下调用fork函数创建子进程,在所述子进程中调用execve函数执行所述执行请求。

[0008] 在一些实施方式中,方法还包括:在所述当前进程和所述子进程间设置无名管道,并进行重定向以接收所述execve函数执行的状态及结果。

[0009] 在一些实施方式中,方法还包括:动态申请内存空间保存所述执行的状态及结果,并将所述内存空间的地址写入所述命令执行管理阵列。

[0010] 在一些实施方式中,所述接收返回的执行状态及结果包括:判断所述返回的执行状态及结果是否属于所述专属类型;以及响应于所述返回的执行状态及结果不属于所述专属类型,将所述返回的执行状态及结果返回原地址。

[0011] 在一些实施方式中,方法还包括:根据当前设备型号或场景判断是否需要创建命令执行管理阵列和/或消息队列;以及响应于需要创建命令执行管理阵列和/或消息队列,发送使能信号并创建命令处理线程。

[0012] 本发明实施例的另一方面,提供了一种管理控制设备的系统,包括:创建模块,配置用于申请内存空间以设置命令执行管理阵列,并设置所述命令执行管理阵列的最大信息长度,申请消息队列资源以创建消息队列并定义所述消息队列发送数据的专属类型;判断模块,配置用于响应于接收到控制主机信息,将所述控制主机信息加入所述命令执行管理阵列,并判断所述控制主机信息的长度是否小于或等于所述命令执行管理阵列的最大信息长度;消息模块,配置用于响应于所述控制主机信息的长度小于或等于所述命令执行管理阵列的最大信息长度,通过所述消息队列使用所述专属类型发送执行请求并接收返回的执行状态及结果;以及初始化模块,配置用于通知主机所述控制主机信息已经执行完成,并响应于所述主机获取所述执行状态及结果,初始化所述命令执行管理阵列和所述消息队列。

[0013] 本发明实施例的又一方面,还提供了一种计算机设备,包括:至少一个处理器;以及存储器,所述存储器存储有可在所述处理器上运行的计算机指令,所述指令由所述处理器执行时实现如上方法的步骤。

[0014] 本发明实施例的再一方面,还提供了一种计算机可读存储介质,计算机可读存储介质存储有被处理器执行时实现如上方法步骤的计算机程序。

[0015] 本发明具有以下有益技术效果:采用单线程方案逐层进行处理,减少了控制数据流对业务数据流的影响,然后通过线程间通信机制实现各层间控制数据的交流,以实现对于Linux系统设备的管理,在避免对业务产生影响的同时,实现对设备的管理和操控。

## 附图说明

[0016] 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,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的前提下,还可以根据这些附图获得其他的实施例。

[0017] 图1为本发明提供的管理控制设备的方法的实施例的示意图;

[0018] 图2为本发明提供的管理控制设备的计算机设备的实施例的硬件结构示意图;

[0019] 图3为本发明提供的管理控制设备的计算机存储介质的实施例的示意图。

## 具体实施方式

[0020] 为使本发明的目的、技术方案和优点更加清楚明白,以下结合具体实施例,并参照附图,对本发明实施例进一步详细说明。

[0021] 需要说明的是,本发明实施例中所有使用“第一”和“第二”的表述均是为了区分两

个相同名称非相同的实体或者非相同的参量,可见“第一”“第二”仅为了表述的方便,不应理解为对本发明实施例的限定,后续实施例对此不再一一说明。

[0022] 基于上述目的,本发明实施例的第一个方面,提出了一种管理控制设备的方法的实施例。图1示出的是本发明提供的管理控制设备的方法的实施例的示意图。如图1所示,本发明实施例包括如下步骤:

[0023] S1、申请内存空间以设置命令执行管理阵列,并设置所述命令执行管理阵列的最大信息长度,申请消息队列资源以创建消息队列并定义所述消息队列发送数据的专属类型;

[0024] S2、响应于接收到控制主机信息,将所述控制主机信息加入所述命令执行管理阵列,并判断所述控制主机信息的长度是否小于或等于所述命令执行管理阵列的最大信息长度;

[0025] S3、响应于所述控制主机信息的长度小于或等于所述命令执行管理阵列的最大信息长度,通过所述消息队列使用所述专属类型发送执行请求并接收返回的执行状态及结果;以及

[0026] S4、通知主机所述控制主机信息已经执行完成,并响应于所述主机获取所述执行状态及结果,初始化所述命令执行管理阵列和所述消息队列。

[0027] 本发明实施例采用C高级语言进行控制机制的描述,使用通用处理器CPU作为主机端运行主程序,在某一单个模块中启用一独立线程完成本控制功能的处理,而在整个系统中,通过多线程处理机制完成控制数据在各模块的封装或处理,再由线程间通信的方式如消息队列进行各模块的数据交互,从而完成对Linux系统的管理和控制。使用此控制管理机制,能够在避免对原有工作机制影响的情况下,实现更为多元化的系统控制机制,满足对多行业、多场景的需求,提高产品适应性。

[0028] 在一些实施方式中,方法还包括:根据当前设备型号或场景判断是否需要创建命令执行管理阵列和/或消息队列;以及响应于需要创建命令执行管理阵列和/或消息队列,发送使能信号并创建命令处理线程。判断当前设备型号或场景是否需要提供此服务机制,若不需要,则此功能不使能,降低CPU负担,若需要则进行功能使能,通过原有信息交互通道发送一使能信息,启用功能,创建相应机制。

[0029] 申请内存空间以设置命令执行管理阵列,并设置所述命令执行管理阵列的最大信息长度,申请消息队列资源以创建消息队列并定义所述消息队列发送数据的专属类型。申请内存空间,设置命令执行管理阵列,限制命令执行管理阵列的长度,最大信息长度可根据实际情况进行设定,由平台交互模块负责维护;申请消息队列资源,创建专属消息队列及定义消息队列发送数据专属类型。

[0030] 在一些实施方式中,方法还包括:响应于所述命令执行管理阵列设置完成,启用新线程监听所述消息队列以检测是否存在执行请求。通过原有信息交互通道告知机箱控制模块,启用新线程监听执行请求。

[0031] 响应于接收到控制主机信息,将所述控制主机信息加入所述命令执行管理阵列,并判断所述控制主机信息的长度是否小于或等于所述命令执行管理阵列的最大信息长度。被控设备平台收到控制主机信息后,先对控制主机信息进行解析、验证,验证方式需根据实际应用场景来设置,例如指令是否支持,验证未通过则给主机端返回相应错误;验证通过则

将控制主机信息添加到命令执行管理阵列。组装shell执行命令字符串,并校验长度合法性,也即是判断命令字符串的长度是否小于或等于命令执行管理阵列的最大信息长度。

[0032] 响应于所述控制主机信息的长度小于或等于所述命令执行管理阵列的最大信息长度,通过所述消息队列使用所述专属类型发送执行请求并接收返回的执行状态及结果。

[0033] 在一些实施方式中,方法还包括:响应于接收到所述执行请求,在当前进程下调用fork函数创建子进程,在所述子进程中调用execve函数执行所述执行请求。Fork()函数是一个系统调用,用于创建进程,创建的这个进程与原来进程几乎完全相同,这个新产生的进程称为子进程,原来的进程称为父进程。一个进程调用fork()函数后,系统先给新的进程分配资源,例如存储数据和代码的空间,然后把原来的集成的所有值都复制到新的进程中,只有少数值与原来的进程的值不同,相当于克隆了一个原来的进程。需要注意的一点就是:调用fork函数之后,父进程和子进程同时执行的代码段是fork函数之后的代码,而之前的代码已经由父进程执行完毕。

[0034] 在fork函数执行完毕后,如果创建新进程成功,则出现两个进程,一个是子进程,一个是父进程。在子进程中,fork函数返回0,在父进程中,fork函数返回新创建的子进程的进程ID(标识),可以通过fork返回的值来判断当前进程是子进程还是父进程。Fork出错可能有两种原因,第一、当前的进程数已经达到了系统规定的上限,这时errno的值被设置为EAGAIN;第二、系统内存不足,这时errno的值被设置为ENOMEM。创建子进程成功后,系统中出现两个基本完全相同的进程,这两个进程执行没有固定的先后顺序,需要根据系统的进程调度策略决定进程执行的顺序。每个进程都有一个独特的进程标识符,可以通过getpid()函数获得,还有一个记录父进程pid的变量,可以通过getppid()函数获得变量的值。

[0035] Exec函数族提供一个在进程中启动另一个程序的方法,它可以根据执行的文件名或目录名找到可执行文件,并用它来取代原调用进程的数据段、代码段和堆栈段,在执行完之后,原调用进程的内容除了进程号之外,其他全部被新程序的内容替换了。进程调用一种exec函数时,该进程完全由新程序替换,而新程序则从其main函数开始执行,因而调用exec并不创建新的进程,所以前后的进程ID(也包括父进程号、进程组号、当前工作目录……)并未改变。Exec只是用另一个新程序替换了当前进程的正文、数据、堆和栈段。Execve的第五位为v表示参数传递为构造指针数组方式,第六位为e表示可传递新进程环境变量。

[0036] 在一些实施方式中,方法还包括:在所述当前进程和所述子进程间设置无名管道,并进行重定向以接收所述execve函数执行的状态及结果。

[0037] 管道是Linux从unix继承过来的进程间通信机制,思想是在内核中创建一个共享文件(该文件是一种伪文件,不占用磁盘大小,是一个缓冲区),使得通信双方利用这个共享文件实现通信,这个传递消息的共享文件就被称为“管道”。无名管道在文件系统中没有文件节点,且能用于具有亲缘关系的进程间通信。无名管道可以用pipe()函数进行创建。重定向就是把一些命令输出的错误或正确的信息放到指定的文件中,这样比较好管理。

[0038] 在一些实施方式中,所述接收返回的执行状态及结果包括:判断所述返回的执行状态及结果是否属于所述专属类型;以及响应于所述返回的执行状态及结果不属于所述专属类型,将所述返回的执行状态及结果返回原地址。

[0039] 通知主机所述控制主机信息已经执行完成,并响应于所述主机获取所述执行状态及结果,初始化所述命令执行管理阵列和所述消息队列。

[0040] 在一些实施方式中,方法还包括:动态申请内存空间保存所述执行的状态及结果,并将所述内存空间的地址写入所述命令执行管理阵列。平台交互模块动态申请空间,保存执行状态及结果,并将内存地址放于对应命令执行管理阵列中,以便维护和取用。主机取走结果后,释放存储该结果所用空间,并初始化命令管理矩阵及消息队列。

[0041] 本发明采用单线程方案逐层进行处理,减少了控制数据流对业务数据流的影响,然后通过线程间通信机制实现各层间控制数据的交流,可以提升控制灵活性及降低控制数据流对其他性能的影响,扩展存储设备控制方式,拓宽应用场景,提高存储设备的竞争力,在避免对业务产生影响的同时,实现对设备的管理和操控。

[0042] 需要特别指出的是,上述管理控制设备的方法的各个实施例中的各个步骤均可以相互交叉、替换、增加、删减,因此,这些合理的排列组合变换之于管理控制设备的方法也应当属于本发明的保护范围,并且不应将本发明的保护范围局限在实施例之上。

[0043] 基于上述目的,本发明实施例的第二个方面,提出了一种管理控制设备的系统,包括:创建模块,配置用于申请内存空间以设置命令执行管理阵列,并设置所述命令执行管理阵列的最大信息长度,申请消息队列资源以创建消息队列并定义所述消息队列发送数据的专属类型;判断模块,配置用于响应于接收到控制主机信息,将所述控制主机信息加入所述命令执行管理阵列,并判断所述控制主机信息的长度是否小于或等于所述命令执行管理阵列的最大信息长度;消息模块,配置用于响应于所述控制主机信息的长度小于或等于所述命令执行管理阵列的最大信息长度,通过所述消息队列使用所述专属类型发送执行请求并接收返回的执行状态及结果;以及初始化模块,配置用于通知主机所述控制主机信息已经执行完成,并响应于所述主机获取所述执行状态及结果,初始化所述命令执行管理阵列和所述消息队列。

[0044] 在一些实施方式中,系统还包括监听模块,配置用于:响应于所述命令执行管理阵列设置完成,启用新线程监听所述消息队列以检测是否存在执行请求。

[0045] 在一些实施方式中,系统还包括第二创建模块,配置用于:响应于接收到所述执行请求,在当前进程下调用fork函数创建子进程,在所述子进程中调用execve函数执行所述执行请求。

[0046] 在一些实施方式中,系统还包括接收模块,配置用于:在所述当前进程和所述子进程间设置无名管道,并进行重定向以接收所述execve函数执行的状态及结果。

[0047] 在一些实施方式中,系统还包括存储模块,配置用于:动态申请内存空间保存所述执行的状态及结果,并将所述内存空间的地址写入所述命令执行管理阵列。

[0048] 在一些实施方式中,所述消息模块配置用于:判断所述返回的执行状态及结果是否属于所述专属类型;以及响应于所述返回的执行状态及结果不属于所述专属类型,将所述返回的执行状态及结果返回原地址。

[0049] 在一些实施方式中,系统还包括第二判断模块,配置用于:根据当前设备型号或场景判断是否需要创建命令执行管理阵列和/或消息队列;以及响应于需要创建命令执行管理阵列和/或消息队列,发送使能信号并创建命令处理线程。

[0050] 本发明采用单线程方案逐层进行处理,减少了控制数据流对业务数据流的影响,然后通过线程间通信机制实现各层间控制数据的交流,可以提升控制灵活性及降低控制数据流对其他性能的影响,扩展存储设备控制方式,拓宽应用场景,提高存储设备的竞争力,

在避免对业务产生影响的同时,实现对设备的管理和操控。

[0051] 基于上述目的,本发明实施例的第三个方面,提出了一种计算机设备,包括:至少一个处理器;以及存储器,存储器存储有可在处理器上运行的计算机指令,指令由处理器执行以实现如下步骤:S1、申请内存空间以设置命令执行管理阵列,并设置所述命令执行管理阵列的最大信息长度,申请消息队列资源以创建消息队列并定义所述消息队列发送数据的专属类型;S2、响应于接收到控制主机信息,将所述控制主机信息加入所述命令执行管理阵列,并判断所述控制主机信息的长度是否小于或等于所述命令执行管理阵列的最大信息长度;S3、响应于所述控制主机信息的长度小于或等于所述命令执行管理阵列的最大信息长度,通过所述消息队列使用所述专属类型发送执行请求并接收返回的执行状态及结果;以及S4、通知主机所述控制主机信息已经执行完成,并响应于所述主机获取所述执行状态及结果,初始化所述命令执行管理阵列和所述消息队列。

[0052] 在一些实施方式中,步骤还包括:响应于所述命令执行管理阵列设置完成,启用新线程监听所述消息队列以检测是否存在执行请求。

[0053] 在一些实施方式中,步骤还包括:响应于接收到所述执行请求,在当前进程下调用fork函数创建子进程,在所述子进程中调用execve函数执行所述执行请求。

[0054] 在一些实施方式中,步骤还包括:在所述当前进程和所述子进程间设置无名管道,并进行重定向以接收所述execve函数执行的状态及结果。

[0055] 在一些实施方式中,步骤还包括:动态申请内存空间保存所述执行的状态及结果,并将所述内存空间的地址写入所述命令执行管理阵列。

[0056] 在一些实施方式中,所述接收返回的执行状态及结果包括:判断所述返回的执行状态及结果是否属于所述专属类型;以及响应于所述返回的执行状态及结果不属于所述专属类型,将所述返回的执行状态及结果返回原地址。

[0057] 在一些实施方式中,步骤还包括:根据当前设备型号或场景判断是否需要创建命令执行管理阵列和/或消息队列;以及响应于需要创建命令执行管理阵列和/或消息队列,发送使能信号并创建命令处理线程。

[0058] 如图2所示,为本发明提供的上述管理控制设备的计算机设备的一个实施例的硬件结构示意图。

[0059] 以如图2所示的装置为例,在该装置中包括一个处理器201以及一个存储器202,并还可以包括:输入装置203和输出装置204。

[0060] 处理器201、存储器202、输入装置203和输出装置204可以通过总线或者其他方式连接,图2中以通过总线连接为例。

[0061] 存储器202作为一种非易失性计算机可读存储介质,可用于存储非易失性软件程序、非易失性计算机可执行程序以及模块,如本申请实施例中的管理控制设备的方法对应的程序指令/模块。处理器201通过运行存储在存储器202中的非易失性软件程序、指令以及模块,从而执行服务器的各种功能应用以及数据处理,即实现上述方法实施例的管理控制设备的方法。

[0062] 存储器202可以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、至少一个功能所需要的应用程序;存储数据区可存储根据管理控制设备的方法的使用所创建的数据等。此外,存储器202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、或其他非易失性固态存储器件。在一些实施例中,存储器202可选包括相对于处理器201远程设置的存储器,这些远程存储器可以通过网络连接至本地模块。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0063] 输入装置203可接收输入的用户名和密码等信息。输出装置204可包括显示屏等显示设备。

[0064] 一个或者多个管理控制设备的方法对应的程序指令/模块存储在存储器202中,当被处理器201执行时,执行上述任意方法实施例中的管理控制设备的方法。

[0065] 执行上述管理控制设备的方法的计算机设备的任何一个实施例,可以达到与之对应的前述任意方法实施例相同或者相类似的效果。

[0066] 本发明还提供了一种计算机可读存储介质,计算机可读存储介质存储有被处理器执行时执行如上方法的计算机程序。

[0067] 如图3所示,为本发明提供的上述管理控制设备的计算机存储介质的一个实施例的示意图。以如图3所示的计算机存储介质为例,计算机可读存储介质3存储有被处理器执行时执行如上方法的计算机程序31。

[0068] 最后需要说明的是,本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,可以通过计算机程序来指令相关硬件来完成,管理控制设备的方法的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,程序的存储介质可为磁碟、光盘、只读存储记忆体(ROM)或随机存储记忆体(RAM)等。上述计算机程序的实施例,可以达到与之对应的前述任意方法实施例相同或者相类似的效果。

[0069] 以上是本发明公开的示例性实施例,但是应当注意,在不背离权利要求限定的本发明实施例公开的范围的前提下,可以进行多种改变和修改。根据这里描述的公开实施例的方法权利要求的功能、步骤和/或动作不需以任何特定顺序执行。此外,尽管本发明实施例公开的元素可以以个体形式描述或要求,但除非明确限制为单数,也可以理解为多个。

[0070] 应当理解的是,在本文中使用的,除非上下文清楚地支持例外情况,单数形式“一个”旨在也包括复数形式。还应当理解的是,在本文中使用的“和/或”是指包括一个或者一个以上相关联地列出的项目的任意和所有可能组合。

[0071] 上述本发明实施例公开实施例序号仅仅为了描述,不代表实施例的优劣。

[0072] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成,也可以通过程序来指令相关的硬件完成,程序可以存储于一种计算机可读存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0073] 所属领域的普通技术人员应当理解:以上任何实施例的讨论仅为示例性的,并非旨在暗示本发明实施例公开的范围(包括权利要求)被限于这些例子;在本发明实施例的思路下,以上实施例或者不同实施例中的技术特征之间也可以进行组合,并存在如上的本发明实施例的不同方面的许多其它变化,为了简明它们没有在细节中提供。因此,凡在本发明实施例的精神和原则之内,所做的任何省略、修改、等同替换、改进等,均应包含在本发明实施例的保护范围之内。

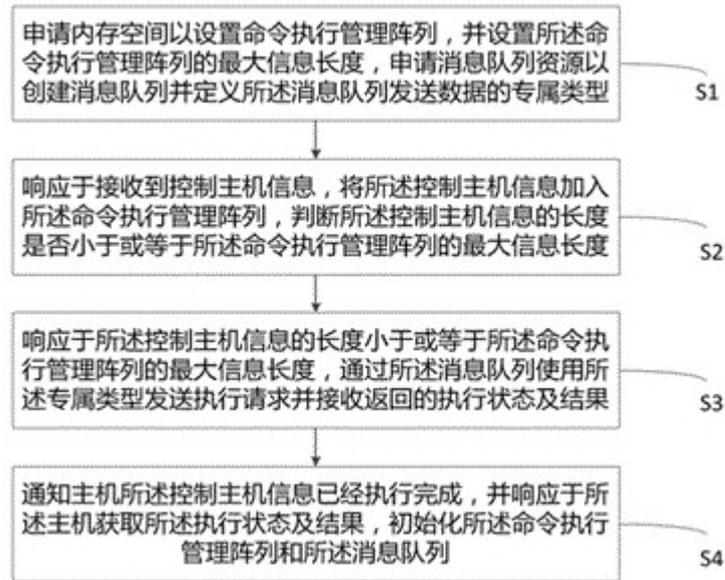


图1

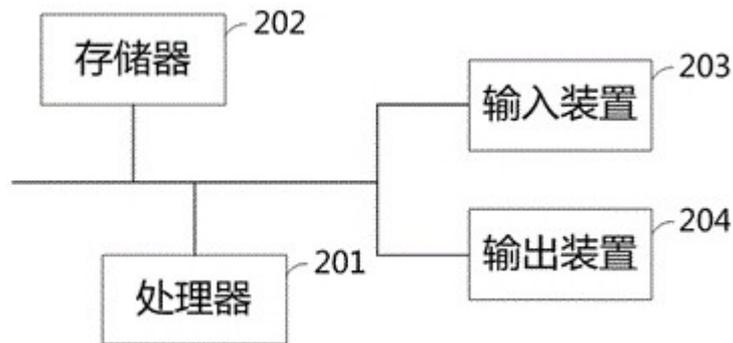


图2

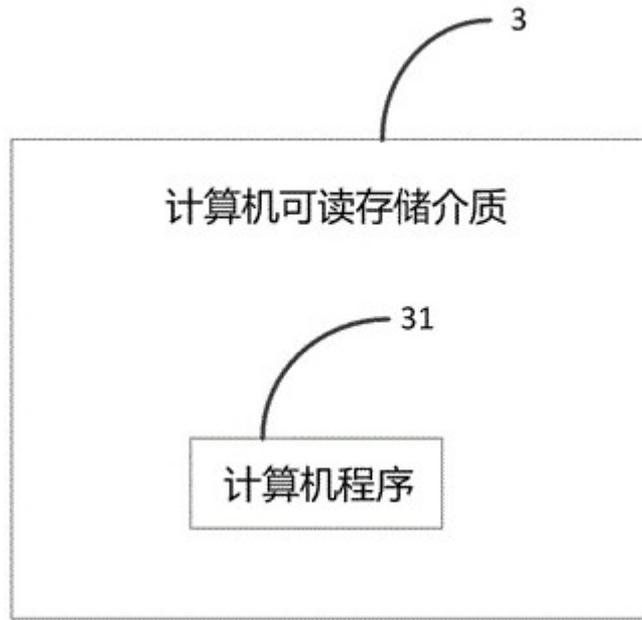


图3