



(12)发明专利

(10)授权公告号 CN 103377078 B

(45)授权公告日 2017.04.12

(21)申请号 201210105512.5

(22)申请日 2012.04.11

(65)同一申请的已公布的文献号
申请公布号 CN 103377078 A

(43)申请公布日 2013.10.30

(73)专利权人 广州地铁集团有限公司
地址 510335 广东省广州市海珠区新港东
路618号南丰汇环球展贸中心第11至
14层
专利权人 中国铁道科学研究院

(72)发明人 丁建隆 何霖 蔡昌俊 李晋
肖宝弟 贾学祥 郇洪民 段晨宁
尹逊政 李博 王芑 陈宁宁
李亮 徐杰 吕利民 张楚潘
黎晓东

(74)专利代理机构 广州新诺专利商标事务所有
限公司 44100

代理人 罗毅萍

(51)Int.Cl.
G06F 9/48(2006.01)

(56)对比文件
CN 101290588 A,2008.10.22,
CN 101290588 A,2008.10.22,
CN 1949178 A,2007.04.18,
US 5274823 A,1993.12.28,

审查员 董泽华

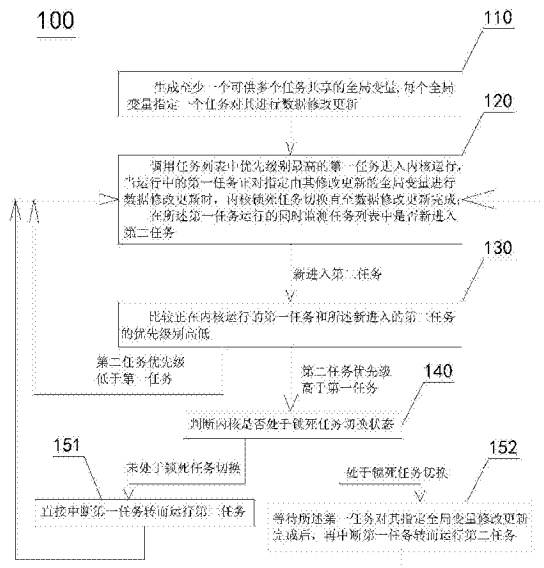
权利要求书2页 说明书8页 附图4页

(54)发明名称

车载ATP实时任务调度方法及其任务调度系统

(57)摘要

本发明属于城市轨道交通CBTC技术领域,具体公开了一种车载ATP实时任务调度方法及其任务调度系统。本发明方法包括生成全局变量、调用任务列表中优先级别最高的第一任务进入内核运行、比较正在内核运行的第一任务和所述新进入的第二任务的优先级别高低、继续按照优先级别高低依次调用任务列表中的各个任务进入内核运行或者中断第一任务转而运行第二任务。本发明系统包括全局变量生成模块、内核运行模块、优先级比较模块、以及内核任务运行状态判断模块。本发明主要是减少任务间的通信等待时间、提高切换效率、以及提高实时防护安全性能,以满足CBTC系统车载ATP子系统多变量、多任务、强实时的控制任务需要。



1. 一种车载ATP实时任务调度方法,其特征在于,包括:

生成至少一个可供多个任务共享的全局变量,每个全局变量指定一个任务对其进行数据修改更新;

调用任务列表中优先级别最高的第一任务进入内核运行,当运行中的第一任务正对指定由其进行数据修改更新的全局变量进行数据修改更新时,内核锁死任务切换直至数据修改更新完成,运行中的第一任务可使用指定由其他任务负责数据修改更新的全局变量;在所述第一任务运行的同时监测任务列表中是否新进入第二任务;

当新进入第二任务后,比较正在在内核运行的第一任务和所述新进入的第二任务的优先级别高低;

若所述第二任务的优先级低于所述第一任务,则继续按照优先级别高低依次调用任务列表中的各个任务进入内核运行;

若所述第二任务的优先级高于所述第一任务,则判断内核是否处于锁死任务切换状态;

若未处于锁死任务切换状态,则直接中断第一任务转而运行第二任务;

若正处于锁死任务切换状态,则等待内核解除任务切换锁死状态后,再中断第一任务转而运行第二任务。

2. 根据权利要求1所述的车载ATP实时任务调度方法,其特征在于,在所述生成至少一个可供多个任务共享的全局变量之前,该方法还包括:

创建多个任务,并按照各任务之间的关系分配不同的优先级;

激活每个任务以分别执行各任务相应的程序代码,并在任务定时计数完成后进入到任务表等待内核调用运行。

3. 根据权利要求1所述的车载ATP实时任务调度方法,其特征在于,在生成至少一个可供多个任务共享的全局变量之前,该方法还包括:

激活一个硬件定时器,该硬件定时器按照设定的周期递增以构建一个时间参考轴;

在各个全局变量生成时或被修改更新后,立即读取所述时间参考轴上的时间数值作为该全局变量的时间戳;

在各个任务被内核调用运行时也从所述时间参考轴上读取一个时间数值作为本任务的时间戳;

在各个任务调用全局变量时,先获取本任务的时间戳与所调用的全局变量的时间戳的时间差值,再比较该时间差值与预设的时间阈值以确定对应全局变量的有效性。

4. 根据权利要求3所述的车载ATP实时任务调度方法,其特征在于,所述比较该时间差值与预设的时间阈值以确定对应全局变量的有效性,具体是:

当该时间差值小于或者等于预设的时间阈值时,认定该全局变量有效;

当该时间差值大于预设的时间阈值时,认定该全局变量无效。

5. 根据权利要求1所述的车载ATP实时任务调度方法,其特征在于,运行中的第一任务使用指定由其他任务负责数据修改更新的全局变量的具体方式是:

在该第一任务的堆栈中定义一个映射的局部变量以供使用,其中该局部变量与指定由该第一任务负责数据修改更新的全局变量相对应。

6. 一种车载ATP实时任务调度系统,其特征在于,包括:

全局变量生成模块,用于生成至少一个可供多个任务共享的全局变量,每个全局变量指定一个任务对其进行数据修改更新;

内核运行模块,用于调用任务列表中优先级别最高的第一任务进入内核运行,当运行中的第一任务正对指定由其进行数据修改更新的全局变量进行数据修改更新时,内核锁死任务切换直至数据修改更新完成,运行中的第一任务可使用指定由其他任务负责数据修改更新的全局变量;在所述第一任务运行的同时监测任务列表中是否新进入第二任务;

优先级比较模块,当新进入第二任务后,用于比较正在在内核运行的第一任务和所述新进入的第二任务的优先级别高低;若所述第二任务的优先级低于所述第一任务,则指令任务执行模块继续按照优先级别高低依次调用任务列表中的各个任务进入内核运行;若所述第二任务的优先级高于所述第一任务,则启动内核运行状态判断模块;

内核任务运行状态判断模块,用于判断内核是否处于锁死任务切换状态;若未处于锁死任务切换状态,则指令内核运行模块直接中断第一任务转而运行第二任务;若正处于锁死任务切换状态,则指令内核运行模块等待内核解除任务切换锁死状态后,再中断第一任务转而运行第二任务。

7. 根据权利要求6所述的车载ATP实时任务调度系统,其特征在于,该系统还包括:
任务创建模块,用于创建多个任务,并按照各任务之间的关系分配不同的优先级;
任务激活模块,用于激活每个任务以分别执行各任务相应的程序代码,并在任务定时计数完成后进入到任务表等待内核调用运行。

8. 根据权利要求6所述的车载ATP实时任务调度系统,其特征在于,该系统还包括:
硬件定时器,用于按照设定的周期递增以构建一个时间参考轴;
时间戳加载模块,用于在各个全局变量生成时或被修改更新后,立即读取所述时间参考轴上的时间数值作为该全局变量的时间戳;在各个任务被内核调用运行时也从所述时间参考轴上读取一个时间数值作为本任务的时间戳;

时间戳比较模块,用于在各个任务调用全局变量时,先获取本任务的时间戳与所调用的全局变量的时间戳的时间差值,再比较该时间差值与预设的时间阈值以确定对应全局变量的有效性。

9. 根据权利要求8所述的车载ATP实时任务调度系统,其特征在于,该系统的时间戳比较模块中,所述比较该时间差值与预设的时间阈值以确定对应全局变量的有效性,具体是:
当该时间差值小于或者等于预设的时间阈值时,认定该全局变量有效;
当该时间差值大于预设的时间阈值时,认定该全局变量无效。

10. 根据权利要求6所述的车载ATP实时任务调度系统,其特征在于,该系统还包括:
局部变量映射模块,用于当在内核运行模块中的运行第一任务使用指定由其他任务负责数据修改更新的全局变量时,该局部变量映射模块在该第一任务的堆栈中定义一个映射的局部变量以供使用,其中该局部变量与指定由该第一任务负责数据修改更新的全局变量相对应。

车载ATP实时任务调度方法及其任务调度系统

技术领域

[0001] 本发明属于城市轨道交通CBTC技术领域,具体涉及一种基于 $\mu\text{C}/\text{OS-II}$ 嵌入式实时操作系统任务调度机制优化和改进的车载ATP的任务调度方法及其任务调度系统。

背景技术

[0002] CBTC,它是基于无线通信技术的列车自动控制系统。在城市轨道交通信号控制系统中,其车载ATP(列车自动保护装置)负责获取并维护列车位置、接收移动授权并计算安全速度曲线、监督列车状态等功能。这些功能的实现需要车载ATP系统具有极高的实时性,才能对各个任务进行实时的响应和快速的调度。因此车载ATP系统普遍采用嵌入式实时操作系统,例如:TKCG-08型车载ATP装置就采用了 $\mu\text{C}/\text{OS-II}$ 实时操作系统。

[0003] $\mu\text{C}/\text{OS-II}$ 一种基于优先级的开源、抢占式多任务实时操作系统,包含了实时内核、任务管理、时间管理、任务间通信同步(信号量、邮箱、消息、队列)和内存管理等功能。它可以使各个任务独立工作,互不干涉,很容易实现准时而且无误执行,使实时应用程序的设计和扩展变得容易,使应用程序的设计过程大为减化。正是由于这些特点,使得 $\mu\text{C}/\text{OS-II}$ 非常适合使用在实时性、可靠性和安全性要求高的场合。事实上, $\mu\text{C}/\text{OS-II}$ 使用在诸多领域,如军工、航空、航天、医疗设备及工业控制等。

[0004] $\mu\text{C}/\text{OS-II}$ 操作系统属于可剥夺型内核系统,它的功能实现是基于任务的,目前车载ATP的任务调度系统中任务之间的通信与调度主要有以下几种方式:

[0005] (1) 基于信号量的任务管理方式

[0006] $\mu\text{C}/\text{OS-II}$ 中的信号量由两部分组成:一个是信号量的计数值,它是一个16位的无符号整数(0到65,535之间);另一个是由等待该信号量的任务组成的等待任务表。信号量的使用包含建立信号量、等待信号量、查询信号量和发送信号量几个功能。各任务间的通信依靠判断信号量的值来保证任务间数据使用的唯一性。例如任务A与任务B同时使用变量C,与此相关的信号量为D,任务A使用变量前需要通过等待信号量得到信号量D为“有效”的指令才可以正常运行。若信号量无效任务A将进入睡眠态,直到信号量有效或等待超时时间到。

[0007] (2) 基于邮箱的任务管理机制

[0008] 邮箱是 $\mu\text{C}/\text{OS-II}$ 中另一种通讯机制,它可以使一个任务或者中断服务子程序向另一个任务发送一个指针型的变量。该指针指向一个包含了特定“消息”的数据结构。使用邮箱之前,必须先建立该邮箱。该操作可以通过调用OSMboxCreate()函数来完成(见下节),并且要指定指针的初始值。一般情况下,这个初始值是NULL,但也可以初始化一个邮箱,使其在最开始就包含一条消息。如果使用邮箱的目的是用来通知一个事件的发生(发送一条消息),那么就要初始化该邮箱为NULL,因为在开始时,事件还没有发生。如果用户用邮箱来共享某些资源,那么就要初始化该邮箱为一个非NULL的指针。在这种情况下,邮箱被当成一个二值信号量使用。各任务向邮箱中发送控制信息,其他任务使用邮箱关联的信息前,需要从邮箱中获取任务是否可用的授权。若邮箱信息不可用,则任务挂起直到邮箱信息可用或等待超时达到。

[0009] 因此,现有的 $\mu\text{C}/\text{OS-II}$ 系统提供的任务通信与调度机制,虽然可以有效的防护不同任务间同时使用相同的数据,但是有以下缺点:

[0010] (1) 任务间大量的数据交换会造成调度指令非常繁琐,程序结构复杂;

[0011] (2) 单个任务出现异常会造成其他任务死锁;

[0012] (3) 没有专门的数据实时性防护机制,一旦信号量或邮箱工作异常,数据安全无法保障。

发明内容

[0013] 为了解决上述问题,本发明的目的在于提供一种简洁的车载ATP的任务调度方法及其任务调度系统,以减少任务间的通信等待时间和提高切换效率。

[0014] 为实现上述发明目的,本发明所采用技术方案如下:

[0015] 一种车载ATP的任务调度方法,包括:生成至少一个可供多个任务共享的全局变量,每个全局变量指定一个任务对其进行数据修改更新;调用任务列表中优先级别最高的第一任务进入内核运行,当运行中的第一任务正对指定由其进行数据修改更新的全局变量进行数据修改更新时,内核锁死任务切换直至数据修改更新完成;在所述第一任务运行的同时监测任务列表中是否新进入第二任务;当新进入第二任务后,比较正在内核运行的第一任务和所述新进入的第二任务的优先级别高低;若所述第二任务的优先级别低于所述第一任务,则继续按照优先级别高低依次调用任务列表中的各个任务进入内核运行;若所述第二任务的优先级别高于所述第一任务,则判断内核是否处于锁死任务切换状态;若未处于锁死任务切换状态,则直接中断第一任务转而运行第二任务;若正处于锁死任务切换状态,则等待内核解除任务切换锁死状态后,再中断第一任务转而运行第二任务。

[0016] 进一步的,在所述生成至少一个可供多个任务共享的全局变量之前,该方法还包括:创建多个任务,并按照各任务之间的关系分配不同的优先级;激活每个任务以分别执行各任务相应的程序代码,并在任务定时计数完成后进入到任务表等待内核调用运行。

[0017] 进一步的,在生成至少一个可供多个任务共享的全局变量之前,该方法还包括:激活一个硬件定时器,该硬件定时器按照设定的周期递增以构建一个时间参考轴;在各个全局变量生成时或被修改更新后,立即读取所述时间参考轴上的时间数值作为该全局变量的时间戳;在各个任务被内核调用运行时也从所述时间参考轴上读取一个时间数值作为本任务的时间戳;在各个任务调用全局变量时,先获取本任务的时间戳与所调用的全局变量的时间戳的时间差值,再比较该时间差值与预设的时间阈值以确定对应全局变量的有效性。

[0018] 进一步的,所述比较该时间差值与预设的时间阈值以确定对应全局变量的有效性,具体是:当该时间差值小于或者等于预设的时间阈值时,认定该全局变量有效;当该时间差值大于预设的时间阈值时,认定该全局变量无效。

[0019] 进一步的,运行中的第一任务使用指定由其他任务负责数据修改更新的全局变量的具体方式是:在该第一任务的堆栈中定义一个映射的局部变量以供使用,其中该局部变量与指定由该第一任务负责数据修改更新的全局变量相对应。

[0020] 一种车载ATP的任务调度系统,包括:

[0021] 全局变量生成模块,用于生成至少一个可供多个任务共享的全局变量,每个全局变量指定一个任务对其进行数据修改更新;

[0022] 内核运行模块,用于调用任务列表中优先级别最高的第一任务进入内核运行,当运行中的第一任务正对指定由其进行数据修改更新的全局变量进行数据修改更新时,内核锁死任务切换直至数据修改更新完成;在所述第一任务运行的同时监测任务列表中是否新进入第二任务;

[0023] 优先级比较模块,当新进入第二任务后,用于比较正在在内核运行的第一任务和所述新进入的第二任务的优先级别高低;若所述第二任务的优先级低于所述第一任务,则指令任务执行模块继续按照优先级别高低依次调用任务列表中的各个任务进入内核运行;若所述第二任务的优先级高于所述第一任务,则启动内核运行状态判断模块;

[0024] 内核任务运行状态判断模块,用于判断内核是否处于锁死任务切换状态;若未处于锁死任务切换状态,则指令内核运行模块直接中断第一任务转而运行第二任务;若正处于锁死任务切换状态,则指令内核运行模块等待内核解除任务切换锁死状态后,再中断第一任务转而运行第二任务。

[0025] 进一步的,该系统还包括任务创建模块和任务激活模块;其中,任务创建模块,用于创建多个任务,并按照各任务之间的关系分配不同的优先级;其中,任务激活模块,用于激活每个任务以分别执行各任务相应的程序代码,并在任务定时计数完成后进入到任务表等待内核调用运行。

[0026] 进一步的,该系统还包括硬件定时器、时间戳加载模块、以及时间戳比较模块;其中,硬件定时器,用于按照设定的周期递增以构建一个时间参考轴;其中,时间戳加载模块,用于在各个全局变量生成时或被修改更新后,立即读取所述时间参考轴上的时间数值作为该全局变量的时间戳;在各个任务被内核调用运行时也从所述时间参考轴上读取一个时间数值作为本任务的时间戳;其中,时间戳比较模块,用于在各个任务调用全局变量时,先获取本任务的时间戳与所调用的全局变量的时间戳的时间差值,再比较该时间差值与预设的时间阈值以确定对应全局变量的有效性。

[0027] 进一步的,该系统的时间戳比较模块中,所述比较该时间差值与预设的时间阈值以确定对应全局变量的有效性,具体是:当该时间差值小于或者等于预设的时间阈值时,认定该全局变量有效;当该时间差值大于预设的时间阈值时,认定该全局变量无效。

[0028] 进一步的,该系统还包括:局部变量映射模块,用于当在内核运行模块中的运行第一任务使用指定由其他任务负责数据修改更新的全局变量时,该局部变量映射模块在该第一任务的堆栈中定义一个映射的局部变量以供使用,其中该局部变量与指定由该第一任务负责数据修改更新的全局变量相对应。

[0029] 本发明定义了多个全局变量,在任务运行过程中对其对应的全局变量进行修改更新时,内核锁死任务切换不允许其他任务抢占内核,但只要任务未处于锁死任务切换状态(即是运行中的第一任务并不是正在修改更新其管理的全局变量时),比运行中的第一任务优先级高的第二任务就可以随时抢占内核,从而简化了任务调度机制,当然进一步减少了任务间的通信等待时间,自然就提高切换效率,从而可以满足CBTC系统车载ATP子系统多变量、多任务、强实时的控制任务需要。

附图说明

[0030] 此附图说明所提供的图片用来辅助对本发明的进一步理解,构成本申请的一部

分,并不构成对本发明的不当限定,在附图中:

- [0031] 图1是本发明任务调度方法的基本原理流程示意图;
- [0032] 图2是本发明任务调度方法的完整流程示意图;
- [0033] 图3是本发明另一实施例的时间戳加载原理示意图;
- [0034] 图4是图1调度方法对应的调度系统原理框图;
- [0035] 图5是图2调度方法对应的调度系统原理框图;
- [0036] 图6是图3调度方法对应的调度系统原理框图。

具体实施方式

[0037] 下面将结合具体实施方法来详细说明本发明,在本发明的示意性实施及说明用来解释本发明,但并不作为对本发明的限定。

[0038] 图1是本发明任务调度方法的基本原理流程示意图。如图1所示,该调度方法100包括:

[0039] S110,生成至少一个可供多个任务共享的全局变量,每个全局变量指定一个任务对其进行数据修改更新;

[0040] S120,调用任务列表中优先级别最高的第一任务进入内核运行,当运行中的第一任务正对指定由其进行数据修改更新的全局变量进行数据修改更新时,内核锁死任务切换直至数据修改更新完成;在所述第一任务运行的同时监测任务列表中是否新进入第二任务;

[0041] 当新进入第二任务后,则顺序执行S130步骤;

[0042] S130,比较正在在内核运行的第一任务和所述新进入的第二任务的优先级别高低;其比较结果:

[0043] 若所述第二任务的优先级低于所述第一任务,则S120步骤继续执行,也即是继续按照优先级别高低依次调用任务列表中的各个任务进入内核运行;

[0044] 若所述第二任务的优先级高于所述第一任务,则顺序执行S140步骤;

[0045] S140,判断内核是否处于锁死任务切换状态;

[0046] 判断结果若未处于锁死任务切换状态,则顺序执行S151步骤,若正处于锁死任务切换状态,则顺序执行S152步骤;

[0047] S151,直接中断第一任务转而运行第二任务

[0048] S152,等待内核解除任务切换锁死状态后,再中断第一任务转而运行第二任务。

[0049] 根据上述实施例方法知,本发明同样是基于任务优先级的调度机制,但区别在于当一个更高优先级人物到来时,中断调用的方式完全不同。例如任务列表中有五个不同的任务:T1、T2、T3、T4、T5,若T1-T5的分配的优先级依次从高到低,那么正常情况首先调用T1进入内核运行,然后依次调入T2、T3、T4、T5进入内核运行;当新任务Tx进入任务列表时,正在执行任务T3;如果Tx的优先级别小于T3,则继续T3的运行,当然在T3运行完成后,也同样是根据T4、T5、Tx他们三个的优先级别高低依次调入内核运行;但如果Tx的优先级别高于T3,则先判断内核是不是处于锁死任务切换状态,如果内核未处于锁死任务切换状态则直接中断T3转而运行Tx,如果内核未处于锁死任务切换状态则需等待内核解除任务切换锁死状态后,才能中断T3转而运行Tx。

[0050] 所谓的任务锁死切换状态,是指运行中的T3正对指定由其修改更新的全局变量进行修改更新时的状态,一旦修改更新完毕即刻解除任务切换锁死状态。因为,在T3运行过程中,包含很多具体的执行动作,当然也包括数据修改更新动作(所谓的数据修改包括数据写入过程),数据修改更新过程中被其他任务中断执行并读取该数据是非常危险的,所以本实施例就采取了任务锁死切换的策略,防止其他任务在此刻抢占内核;同样,T3运行过程中不仅仅包括数据修改更新过程,更多的时间是在执行其他的动作,如果一直锁死内核不予切换,那么Tx等待时间久较长,完全不能达到紧急事件及时处理的要求,所以本实施例任务锁死切换仅仅在数据修改更新过程,当然如果在T3执行过程中需要多次修改更新其对应的全局变量,则执行多次锁死和解除任务切换状态即可。

[0051] 因此,本实施例通过在数据的更新阶段进行的锁死任务切换操作,可以有效避免数据修改更新过程中被其他任务读取的风险。同时,这种仅在数据维护阶段控制任务调度的方法相对于信号量或邮箱来讲减小了任务相互等待的时间。

[0052] 当然,为了更好地配合实施例引入的任务锁死切换策略,本实施例还采用了共享的全局变量+映射的局部变量的方式,即是:生成至少一个可供多个任务共享的全局变量,每个全局变量指定一个任务对其进行修改更新,其他任务调用该全局变量数据就在其堆栈中定义一个映射的局部变量与共享的全局变量对应;例如:五个不同任务和各自的全局变量(一个任务可以对应多个全局变量,但是一个全局变量只能由一个任务进行更新修改)是:T1—a、T2—b、T3—c、T4—d、T5—(e、f),假如T3运行过程中需要用到变量由a、c、e、f,T3除了在可以直接调用c以外,a、e、f都只能在堆栈中映射一个与它们对应的局部变量a1、e2、f1以供使用。实质是把数据的读取和修改更新分为不同的权限,每个数据由生成它的任务进行维护,该任务直接对共享的全局变量进行修改。其他任务使用该数据则应该在本任务堆栈中定义一个映射的局部变量与共享的全局变量对应。

[0053] 图2是本发明任务调度方法的完整流程示意图。如图2所示,该任务调度方法200与任务调度方法100的区别在于,还包括和现有 $\mu\text{C}/\text{OS-II}$ 系统调度方法相同的常规步骤S210:创建多个任务,并按照各任务之间的关系分配不同的优先级;激活每个任务以分别执行各任务相应的程序代码,并在任务定时计数完成后进入到任务表等待内核调用运行。该任务调度方法200中的步骤220至261(262)与任务调度方法100中的步骤S110至151(152)相同,在此不再累述。因为,很多数据需要在系统运行前采集或生成,这些都是必须要通过创建任务分别的执行,比如:车门开关状态对应定义一个车门状态检测任务,在列车运行前肯定要知道车门的开关状态以供其他任务调用,所以要先该车门状态检测任务相应的程序代码指令相应的硬件以获取开关状态,其他任务(比如车辆启动任务)获取该状态数据后才能根据它作出相应的动作。

[0054] 为了保证数据的时效性以提高系统的安全性能,本实施例还引入了有效的数据时效性检查机制,不只依靠操作系统的调度机制来保证数据有效性。本实施例在图2原理基础上,在此引入了时间戳机制,基于硬件定时器生成各个任务的时间戳,依靠时间戳监测来保证任务工作的时效性,即是:在生成至少一个可供多个任务共享的全局变量之前,激活一个硬件定时器,该硬件定时器按照设定的周期递增以构建一个时间参考轴,该时间周期根据实际的系统需要可以是1毫秒或者几毫秒;在各个全局变量生成时或被修改更新后,立即读取所述时间参考轴上的时间数值作为该全局变量的时间戳;在各个任务被内核调用运行时

也从所述时间参考轴上读取一个时间数值作为本任务的时间戳;在各个任务调用全局变量时,先获取本任务的时间戳与所调用的全局变量的时间戳的时间差值,再比较该时间差值与预设的时间阈值以确定对应全局变量的有效性。具体可以是:当该时间差值小于或者等于预设的时间阈值时,认定该全局变量有效;当该时间差值大于预设的时间阈值时,认定该全局变量无效。

[0055] 具体参考图3所示:

[0056] 在时刻 t_1 生成或更新了全局变量 a 、 b ,那么在时间轴上获取时间戳使得该数据为 t_1+a 、 t_1+b ;

[0057] 在时刻 t_2 生成或更新了全局变量 c 、 d ,那么在时间轴上获取时间戳使得该数据为 t_2+c 、 t_2+d ;

[0058] 在时刻 t_3 运行任务1,同理任务1的时间戳为 t_3 ,假如任务1的运行需要调用数据 a 、 b 、 c 、 d ,那么首先用比较获得它们之间的时间差值:

[0059] $R_1 = t_3 - t_1$ 、 $R_2 = t_3 - t_2$

[0060] 然后将 R_1 与预设的时间阈值 Y_1 进行比较,如果 R_1 小于或者等于预设时间阈值 Y_1 时,认为 a 、 b 有效;

[0061] 再将 R_2 与预设的时间阈值 Y_2 进行比较,如果 R_2 大于预设的时间阈值 Y_2 时,认为 c 、 d 无效。

[0062] 本实施例实质是:引入了独立的硬件定时器作为系统时间轴,各个任务可通过读取硬件定时器的寄存器获取自身时间戳。该时间戳一方面作为本任务负责维护数据的生命起点标志供其他任务检查,同时也是检查其他任务提供给本任务使用数据的时效性依据。这样的机制能够保证即使操作系统出现故障导致任务延时执行、数据更新的延误,也不会影响系统的安全性,相当于给系统增加了一道保险。

[0063] 需要说明的是,其时间阈值是根据实际不同情况设置的不同数值,比如:启动列车任务为任务1,需要调用车门开关状态变量为 a ($a=1$ 表示关, $a=0$ 表示开,时刻不同 a 在不断的变化更新),我们设定阈值为1000毫秒;假如任务1开始运行时刻为 t_3 ,首先获得它们之间的时间差值 $R_2 = t_3 - t_2$,只有 R_2 小于或者等于1000毫秒,才认为 a 数据是有效数据。如果该时间阈值设置太大如10分钟,那么当在10分钟之内开关门多次而状态数据没有更新时,这种异常就非常危险,可能出现车门开着,列车却一直行驶。

[0064] 在现有系统中一旦系统出现故障,很可能导致数据更新延误,数据更新延误较长将带来极大的安全隐患,比如列车到站后车门打开了一段时间,但是数据一直没有更新,还处于之前某一时刻的关闭状态,由于没有时间戳去验证,那么将认为该数据有效,则将直接导致车门没关就列车就开动了。但本实施例加入了时间戳机制后,而且是采用的是独立硬件定时器,可有效验证各数据的有效性,从而进一步提高系统的安全性能。

[0065] 下面将描述与前述方法对应的任务调度系统。

[0066] 图4是图1调度方法对应的调度系统原理框图。如图4所示,本实施例公开一种车载ATP的任务调度系统400,包括全局变量生成模块410、内核运行模块420、优先级比较模块430、内核任务运行状态判断模块440、以及局部变量映射模块450。其中,全局变量生成模块410,用于生成至少一个可供多个任务共享的全局变量,每个全局变量指定一个任务对其进行修改更新;其中,局部变量映射模块450,用于当在内核运行模块中的运行第一任务使用

由其他任务负责修改更新的全局变量时,该局部变量映射模块在该第一任务的堆栈中定义一个映射的局部变量以供使用,其中该局部变量与指定由该第一任务负责数据修改更新的全局变量相对应。其中,内核运行模块420,用于调用任务列表中优先级别最高的第一任务进入内核运行,当运行中的第一任务正对指定由其修改更新的全局变量进行数据修改更新时,内核锁死任务切换直至数据修改更新完成;在所述第一任务运行的同时监测任务列表中是否新进入第二任务;其中,优先级比较模块430,当新进入第二任务后,用于比较正在在内核运行的第一任务和所述新进入的第二任务的优先级别高低;若所述第二任务的优先级别低于所述第一任务,则指令任务执行模块继续按照优先级别高低依次调用任务列表中的各个任务进入内核运行;若所述第二任务的优先级别高于所述第一任务,则启动内核运行状态判断模块;其中,内核任务运行状态判断模块440,用于判断内核是否处于锁死任务切换状态;若未处于锁死任务切换状态,则指令内核运行模块直接中断第一任务转而运行第二任务;若正处于锁死任务切换状态,则指令内核运行模块等待内核解除任务切换锁死状态后,再中断第一任务转而运行第二任务。

[0067] 图5是图2调度方法对应的调度系统原理框图。如图5所示,本调度系统500的与调度系统400区别在于,还包括和现有 $\mu\text{C}/\text{OS-II}$ 系统调度系统相同的常规模块:任务创建模块510和任务激活模块520。该系统中的其他模块530-570都和调度系统400中的模块410-450相同,不再累述。其中,任务创建模块510,用于创建多个任务,并按照各任务之间的关系分配不同的优先级;其中,任务激活模块520,用于激活每个任务以分别执行各任务相应的程序代码,并在任务定时计数完成后进入到任务表等待内核调用运行。

[0068] 图6是图3对应的调度方法对应的调度系统原理框图。如图5所示,本调度系统600的与调度系统500区别在于,还包括硬件定时器608、时间戳加载模块609、时间戳比较模块610。该系统中的其他模块601-607都和调度系统500中的模块510-570相同,不再累述。其中,硬件定时器608,用于按照设定的周期递增以构建一个时间参考轴;其中,时间戳加载模块609,用于在各个全局变量生成时或被修改更新后,立即读取所述时间参考轴上的时间数值作为该全局变量的时间戳;在各个任务被内核调用运行时也从所述时间参考轴上读取一个时间数值作为本任务的时间戳;其中,时间戳比较模块610,用于在各个任务调用全局变量时,先获取本任务的时间戳与所调用的全局变量的时间戳的时间差值,再比较该时间差值与预设的时间阈值以确定对应全局变量的有效性。该系统的时间戳比较模块610中,所述比较该时间差值与预设的时间阈值以确定对应全局变量的有效性,具体是:当该时间差值小于或者等于预设的时间阈值时,认定该全局变量有效;当该时间差值大于预设的时间阈值时,认定该全局变量无效。

[0069] 本技术方案针对车载ATP系统的实际运用提出的1套在 $\mu\text{C}/\text{OS-II}$ 操作系统下进行任务调度的解决方案,其可以简化应用软件和操作系统之间的接口,有利于软件架构设计;还能减小任务之间通信的等待时间,提高了切换效率;其增加任务之间数据交换的实时性防护,可提高系统安全性。

[0070] 综上实施例可知,本发明为了实现更加安全快捷的任务调度,主要是采取了以下两点改进:

[0071] 一、本发明采用简洁的任务调度与通信机制,基于 $\mu\text{C}/\text{OS-II}$ 操作系统实际应用环境的分析,放弃了 $\mu\text{C}/\text{OS-II}$ 系统自带的基于信号量、邮箱等任务调度与通信机制。仅使用了

最基本的基于任务优先级和时间管理的功能,通过对不同任务创建时的优先级和执行周期定义,实现了任务的运行机制确定。同时,通过运行期间实施的全局共享变量任务级自锁变更与局部映射变量调用机制实现任务间的数据交换,从而满足了CBTC系统车载ATP子系统多变量、多任务、强实时的控制任务需要。

[0072] 二、本发明是基于时间戳的数据实时性防护,引入了独立的硬件定时器作为系统时间轴,各个任务可通过读取硬件定时器的寄存器获取自身时间戳。该时间戳一方面作为本任务负责维护数据的生命起点标志供其他任务检查,同时也是检查其他任务提供给本任务使用数据的时效性依据,从而进一步提高了系统的安全性能。

[0073] 以上对本发明实施例所提供的技术方案进行了详细介绍,本文中应用了具体个例对本发明实施例的原理以及实施方式进行了阐述,以上实施例的说明只适用于帮助理解本发明实施例的原理;同时,对于本领域的一般技术人员,依据本发明实施例,在具体实施方式以及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

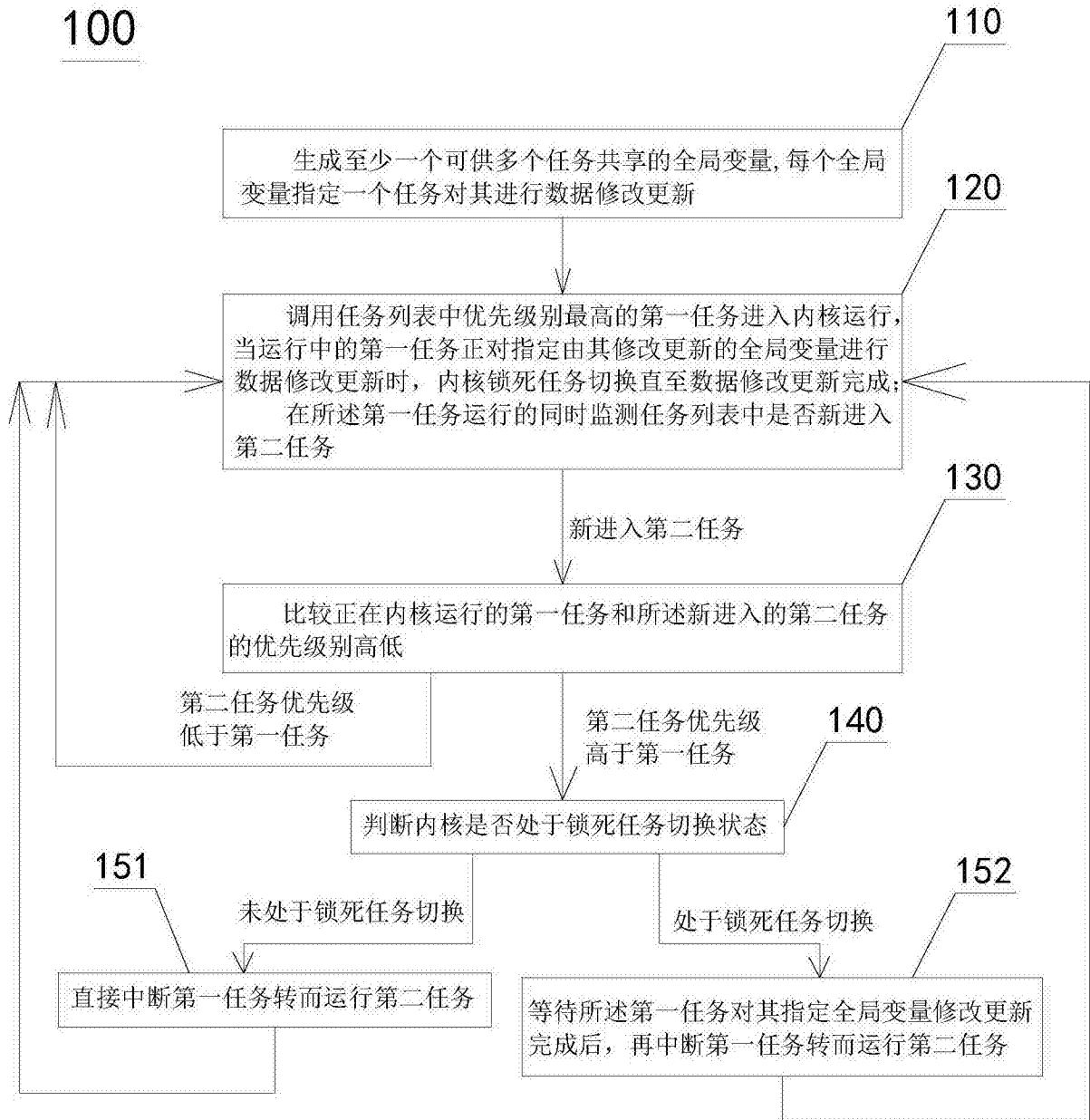


图1

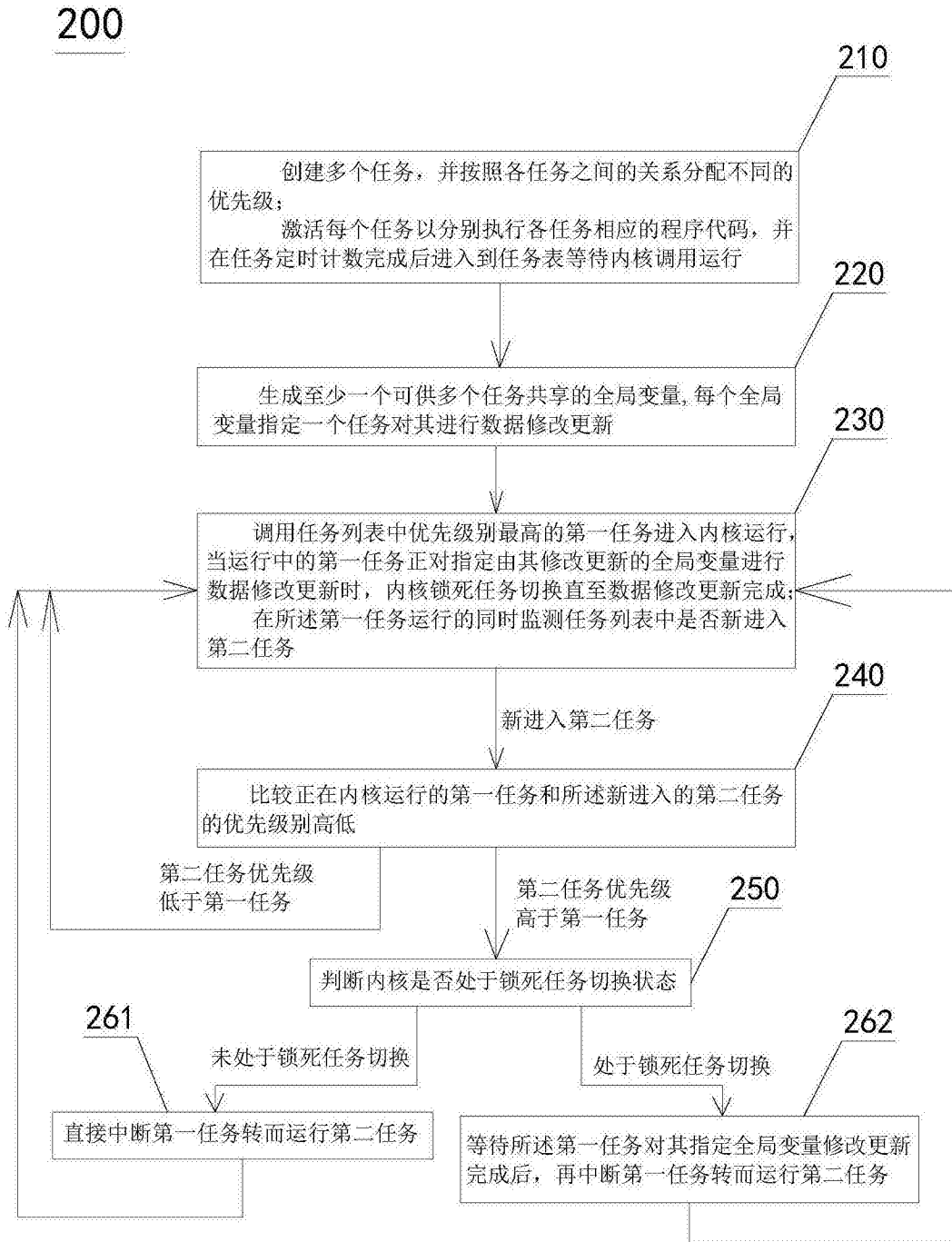


图2

300

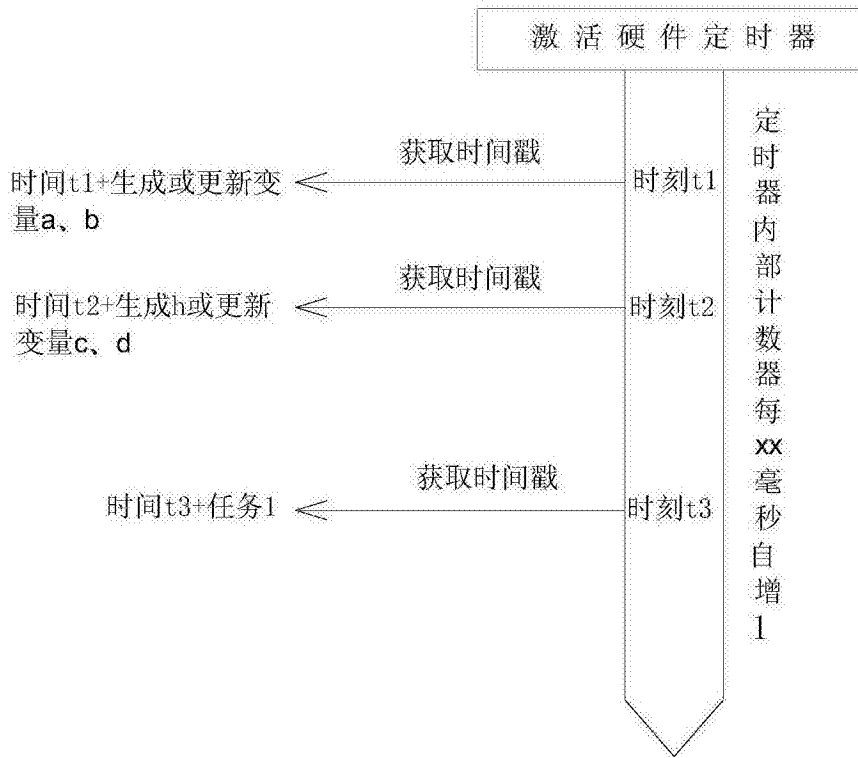


图3



图4



图5

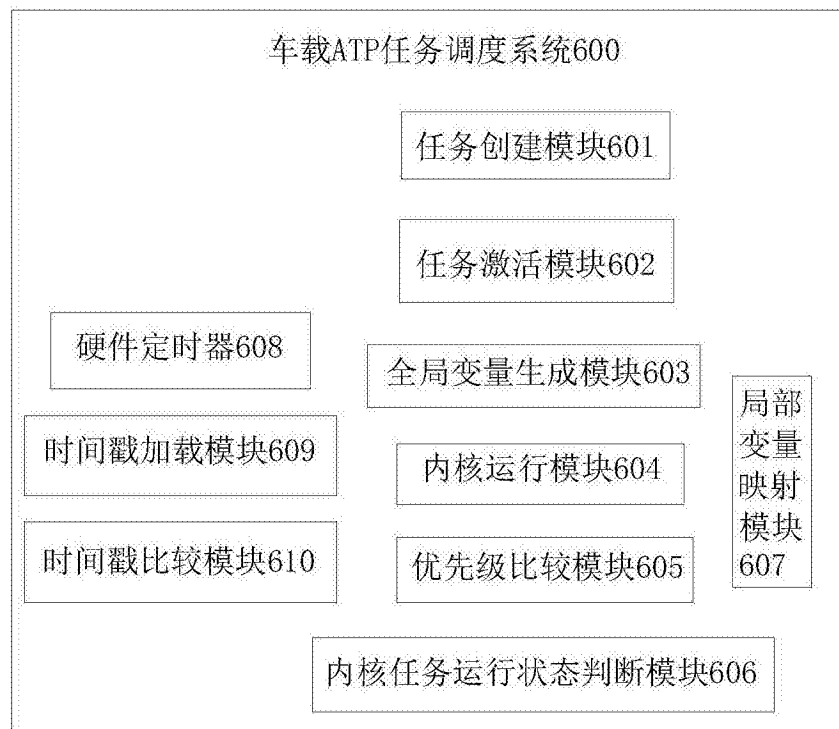


图6