



(12) 发明专利申请

(10) 申请公布号 CN 103473138 A

(43) 申请公布日 2013. 12. 25

(21) 申请号 201310428119. 4

(22) 申请日 2013. 09. 18

(71) 申请人 柳州市博源环科科技有限公司
地址 545006 广西壮族自治区柳州市高新一路 15 号科技工业苑 10 层 1005 号

(72) 发明人 覃祖茂 刘为 袁增伟 杜怡曼
何佳 李东娥 刘晓 黄益农
黄华峰

(74) 专利代理机构 广西南宁汇博专利代理有限公司 45114

代理人 邓晓安

(51) Int. Cl.
G06F 9/50 (2006. 01)

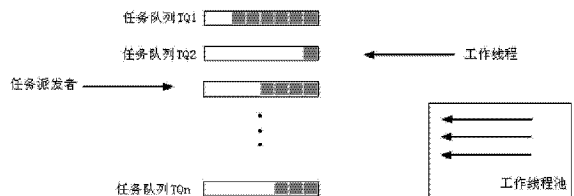
权利要求书1页 说明书4页 附图3页

(54) 发明名称

基于线程池的多任务队列调度方法

(57) 摘要

本发明公开一种基于线程池的多任务队列调度方法,首先根据任务类型创建对应的任务队列,并且任务队列与工作线程池的工作线程没有直接关联;然后创建任务派发者,接收任务并根据任务类型派发到相应的任务队列中;最后当任务派发者派发任务到任务队列中时,再从工作线程池中调用空闲的工作线程执行任务,当任务执行结束后,工作线程回归工作线程池。本发明的方法可以适用于拥有大量不同类型任务且所有任务都需要能够即时并有序处理的情景,不需要为每个任务类型创建一个独立的工作进程或线程,可以极大的有效的节省系统资源且工作线程池内的工作线程可以根据实际情况创建或销毁,使得采用本方法的调度系统拥有良好的线程数可伸缩性。



1. 一种基于线程池的多任务队列调度方法,其特征在于,该调度方法包括以下步骤:
 - (1) 为每种任务类型创建一个对应的任务队列,并分别记为 TQ1, TQ2, ..., TQn;
 - (2) 创建一个或者两个以上任务派发者,接收任务并根据任务类型派发到相应的任务队列中;
 - (3) 初始化工作线程池,空闲状态的工作线程进入睡眠状态;
 - (4) 当任务派发者接收到某一类型的任务 Ti 时,找到其对应的任务队列 TQi,并判断该任务队列的待执行任务是否为空;
 - (5) 如果该任务队列 TQi 的待执行任务为空,则将任务 Ti 放入到该任务队列 TQi 中,并执行以下操作:
 - a. 在工作线程池中查找空闲状态的工作线程,如果工作线程池中无空闲状态的工作线程,则创建一个新的工作线程,并将其加入到工作线程池中;
 - b. 用找到的空闲状态的工作线程或是新建的工作线程来执行任务 Ti;
 - (6) 如果该任务队列 TQi 的待执行任务不为空,则将任务 Ti 放在任务队列 TQi 的末尾,等待任务执行;
 - (7) 当工作线程执行完一个任务 Ti 后,查找 Ti 对应的队列 TQi,然后执行以下操作:
 - a. 将任务 Ti 从 TQi 中移除;
 - b. 如果 TQi 不为空,继续执行队头的任务;
 - c. 如果 TQi 为空,终止执行,工作线程回归工作线程池。
2. 根据权利要求 1 所述的基于线程池的多任务队列调度方法,其特征在于,所述的任务派发者为一个独立的进程或线程。
3. 根据权利要求 1 所述的基于线程池的多任务队列调度方法,其特征在于,所述的工作线程池创建有一个以上的工作线程,且工作线程不与任务队列直接相关联。
4. 根据权利要求 1-3 任一所述的基于线程池的多任务队列调度方法,其特征在于,所述的待执行任务包括未执行任务和正在执行但未完成的任务。
5. 根据权利要求 4 所述的基于线程池的多任务队列调度方法,其特征在于,所述的工作线程回归工作线程池后,工作线程池将销毁该工作线程或是将该工作线程分配给其他任务队列执行任务。

基于线程池的多任务队列调度方法

技术领域

[0001] 本发明属于计算机工程技术领域,涉及了一种基于工作线程池而设计的多任务队列调度方法。

背景技术

[0002] RTOS(Realtime operation system,实时操作系统)是嵌入式应用软件的基础和开发平台,是一个可靠性和可信性很高的实时内核。RTOS 将 CPU 时间、中断、I/O、定时器等资源都包装起来,留给用户一个标准的 API(Application Program Interface,应用程序接口),并根据各个任务的优先级,合理地在不同任务之间分配 CPU 时间。

[0003] RTOS 通常设置于嵌入式系统中实时地完成各种功能,这些嵌入式系统通常具有实时性很好、系统体积小、资源有限和没有图形界面等特点。典型的 RTOS 有:Vxworks、pSOS、Nucleus、eCos、uC/OS-II 等。

[0004] GUI(Graphic User Interface,图形用户接口)系统以图形的方式实现人机交互,调度机制是 GUI 系统的核心,管理 GUI 系统的驱动和各个应用程序,实现各个应用程序对 CPU 资源的共享使用。高性能的 GUI 调度机制不仅占用很少的 CPU 资源,而且能够快速响应用户请求。

[0005] 上述 GUI 调度机制不是指操作系统内核层面的任务调度,而是指具有界面显示的各个应用程序之间的调度机制。例如,开机便进入待机模块,摘机就进入拨号模块,不摘机按功能键就进入功能菜单模块,从功能菜单中选择照相机从而进入拍照模块,此时如果有来电,则进入语音通话模块,在通话完毕挂机后又返回拍照模块。

[0006] 当前典型的 GUI 系统为 Windows 操作系统的 GUI,在嵌入式系统中比较常见的 GUI 系统为 Qtopia 和 MiniGUI。下面分别介绍上述三个 GUI 系统。

[0007] 1. Windows CE(Control System,控制系统)

Windows CE 是嵌入式操作系统,提供完整的 GUI 界面和调度系统,与 PC 版的 Windows 在界面上几乎完全一致。

[0008] 2. MiniGUI

MiniGUI 是针对嵌入式设备的、跨操作系统的图形界面支持系统,属于一种“嵌入式图形中间件”软件产品。在近八年的发展历程中,MiniGUI 已经从一个仅仅是为了在 Linux 显示中文而开发的小技术,发展成为一个跨操作系统的、完善的嵌入式图形用户界面支持系统。MiniGUI 最理想地平台是 Linux 操作系统,也可以支持 uCOSII 等 RTOS。

[0009] 3. Qtopia

Qtopia 由 Qt/Embedded 发展而来,是面向嵌入式系统的 Qt 版本。因为 Qt 是 KDE 等项目使用的 GUI 支持库,因此有许多基于 Qt 的 X Window 程序可以非常方便地移植到 Qtopia 版本上。Qtopia 是一个 C++ 函数库,它提供的控件集风格沿用了 PC 风格。

[0010] 上述现有技术中的 GUI 系统的缺点为:上述 GUI 系统的调度完全使用内核提供的线程调度,每个应用程序不管是否需要界面显示都有独立的线程来接收操作系统的调度。

基于底层强大的操作系统做支撑,各个应用程序的设计虽然具有极大的灵活性,但是,各个应用程序对底层操作系统的依赖太重,通常需要平台化的操作系统的支持,比如 Windows 或者 Linux 操作系统。

[0011] 上述 GUI 系统的资源开销巨大,不能适应功能明确的小型嵌入式系统的应用需求,一般不支持 RTOS 作为底层的操作系统。另外,对于嵌入式应用也不需要复杂的 GUI 系统,过于复杂的 GUI 事实上会降低用户操作的简便性。

[0012] 上述 GUI 系统的各个应用程序之间是一种松散关系,导致需要实现多个应用程序时可能会出现如下的两个问题:

1、如果将要实现的多个应用程序作为不同的应用程序功能模块,则要求这些应用程序功能模块之间不要有太多的相互联系。一旦这些应用程序功能模块之间有复杂的联系,这些应用程序功能模块对系统的资源和设备(比如 Camera、内存)的需求必然会产生冲突,实现起来很难维护;

2、如果将要实现的多个应用程序作为一个单独应用程序功能模块,则会给多个程序员的协同开发带来巨大的维护难度,将会导致多个程序员去开发同一个模块或者一个程序员完成一个规模宏大的模块。

[0013] 而上述系统出现如此的缺点或不足,主要就是其调度方法的逻辑设计不当造成的,因此,我们需要一种逻辑设计更为合理的多任务队列调度方法。

发明内容

[0014] 本发明的目的在于针对现有技术的不足,提供了一种基于操作系统内置的工作线程池以及开发语言提供的队列数据结构而设计的多任务队列调度方法。该调度方法特别适用于拥有大量不同类型任务,且所有任务都需要能够即时并有序处理的情景。本方法的特点在于其拥有良好的进程或线程数可伸缩性;因为其不需要为每个任务类型创建一个独立的工作进程或线程,所以它可以极大的节省系统资源。

[0015] 为了实现上述目的,本发明采取了以下技术方案:

一种基于线程池的多任务队列调度方法,包括以下步骤:

(1) 为每种任务类型创建一个对应的任务队列,并分别记为 TQ1, TQ2, ..., TQn;

(2) 创建一个或者两个以上任务派发者,接收任务并根据任务类型派发到相应的任务队列中;

(3) 初始化工作线程池,空闲状态的工作线程进入睡眠状态;

(4) 当任务派发者接收到某一类型的任务 T_i 时,找到其对应的任务队列 TQi,并判断该任务队列的待执行任务是否为空;

(5) 如果该任务队列 TQi 的待执行任务为空,则将任务 T_i 放入到该任务队列 TQi 中,并执行以下操作:

a. 在工作线程池中查找空闲状态的工作线程,如果工作线程池中无空闲状态的工作线程,则创建一个新的工作线程,并将其加入到工作线程池中;

b. 用找到的空闲状态的工作线程或是新建的工作线程来执行任务 T_i ;

(6) 如果该任务队列 TQi 的待执行任务不为空,则将任务 T_i 放在任务队列 TQi 的末尾,等待任务执行;

(7) 当工作线程执行完一个任务 T_i 后, 查找 T_i 对应的队列 TQ_i , 然后执行以下操作:

- a. 将任务 T_i 从 TQ_i 中移除;
- b. 如果 TQ_i 不为空, 继续执行队头的任务;
- c. 如果 TQ_i 为空, 终止执行, 工作线程回归工作线程池。

[0016] 作为本发明的进一步说明, 以上所述的任务派发者为一个独立的进程或线程。

[0017] 作为本发明的进一步说明, 以上所述的工作线程池创建有一个以上的工作线程, 且工作线程不与任务队列直接相关联。

[0018] 作为本发明的进一步说明, 以上所述的待执行任务包括未执行任务和正在执行但未完成的任务。

[0019] 作为本发明的进一步说明, 以上所述的工作线程回归工作线程池后, 工作线程池将销毁该工作线程或是将该工作线程分配给其他任务队列执行任务。

[0020] 本发明的优点:

1. 在本发明中, 创建有多个任务队列, 这样该方法就可以适用于拥有大量不同类型任务且所有任务都需要能够即时并有序处理的情景。

[0021] 2. 在本发明中, 任务队列与工作线程池的工作线程没有直接关联, 因此, 不需要为每个任务类型创建一个独立的工作进程或线程, 可以极大的有效的节省系统资源。

[0022] 3. 在本发明中, 工作线程池内的工作线程可以根据实际情况创建或销毁, 这样使得采用本方法的调度系统拥有良好的进程或线程数可伸缩性。

[0023] 4. 本发明逻辑设计合理、可行、易实现。

附图说明

[0024] 图 1 是本发明调度方法的示意图。

[0025] 图 2 是本发明中的任务派发者的逻辑设计流程图。

[0026] 图 3 是本发明中的工作线程的逻辑设计流程图。

具体实施方式

[0027] 下面结合附图和实施例对本发明进一步说明。

[0028] 实施例:

如图 1-3 所示, 一种基于线程池的多任务队列调度方法, 包括以下步骤:

(1) 为每种任务类型创建一个对应的任务队列, 并分别记为 TQ_1, TQ_2, \dots, TQ_n ;

(2) 创建一个或者两个以上任务派发者, 接收任务并根据任务类型派发到相应的任务队列中; 所述的任务派发者为一个独立的进程或线程;

(3) 初始化工作线程池, 空闲状态的工作线程进入睡眠状态; 所述的工作线程池创建有一个以上的工作线程, 且工作线程不与任务队列直接相关联;

(4) 当任务派发者接收到某一类型的任务 T_i 时, 找到其对应的任务队列 TQ_i , 并判断该任务队列的待执行任务是否为空; 所述的待执行任务包括未执行任务和正在执行但未完成的任务;

(5) 如果该任务队列 TQ_i 的待执行任务为空, 则将任务 T_i 放入到该任务队列 TQ_i 中, 并执行以下操作:

a. 在工作线程池中查找空闲状态的工作线程,如果工作线程池中无空闲状态的工作线程,则创建一个新的工作线程,并将其加入到工作线程池中;

b. 用找到的空闲状态的工作线程或是新建的工作线程来执行任务 T_i ;

(6)如果该任务队列 TQ_i 的待执行任务不为空,则将任务 T_i 放在任务队列 TQ_i 的末尾,等待任务执行;

(7)当工作线程执行完一个任务 T_i 后,查找 T_i 对应的队列 TQ_i ,然后执行以下操作:

a. 将任务 T_i 从 TQ_i 中移除;

b. 如果 TQ_i 不为空,继续执行队头的任务;

c. 如果 TQ_i 为空,终止执行,工作线程回归工作线程池;所述的工作线程回归工作线程池后,工作线程池将销毁该工作线程或是将该工作线程分配给其他任务队列执行任务。

[0029] 该方法可以适用于拥有大量不同类型任务且所有任务都需要能够即时并有序处理的情景,而且不需要为每个任务类型创建一个独立的工作进程或线程,可以极大的有效的节省系统资源。

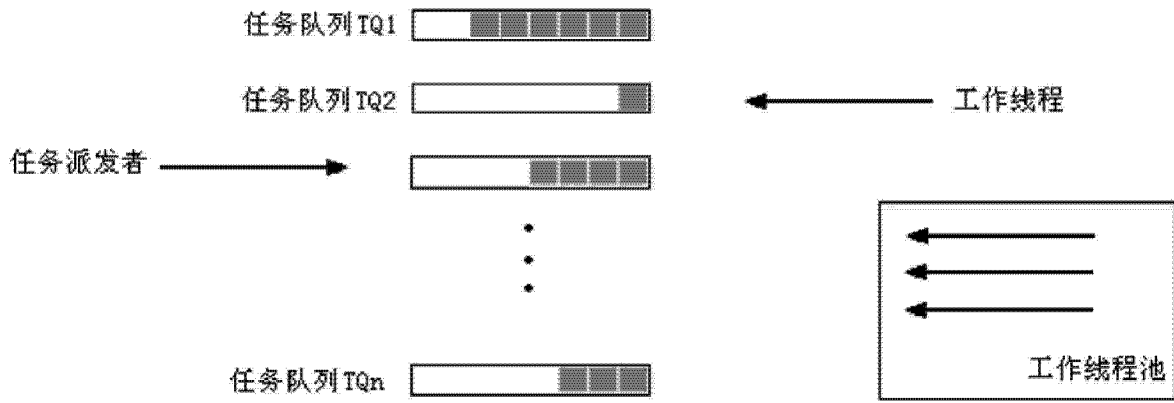


图 1

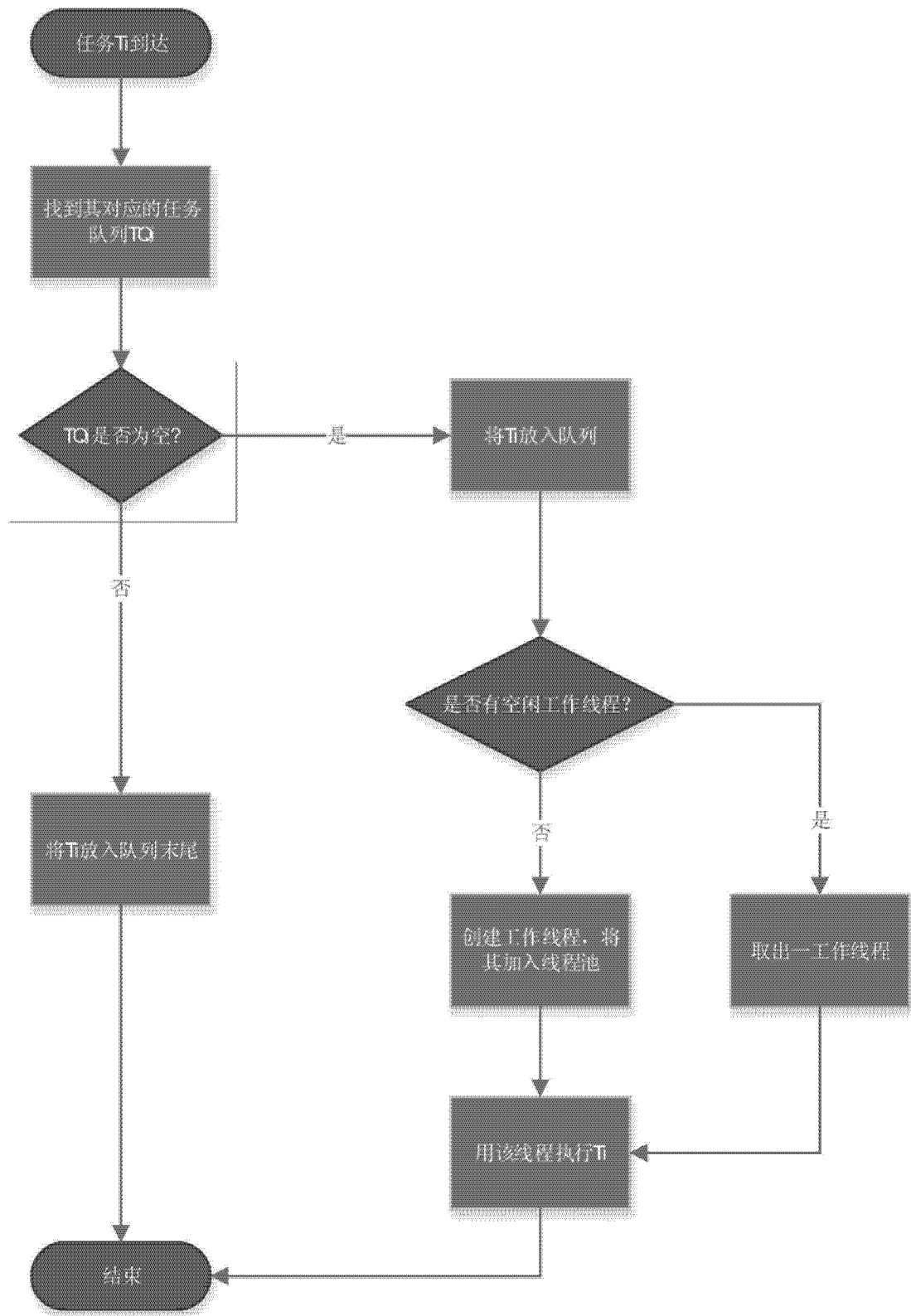


图 2

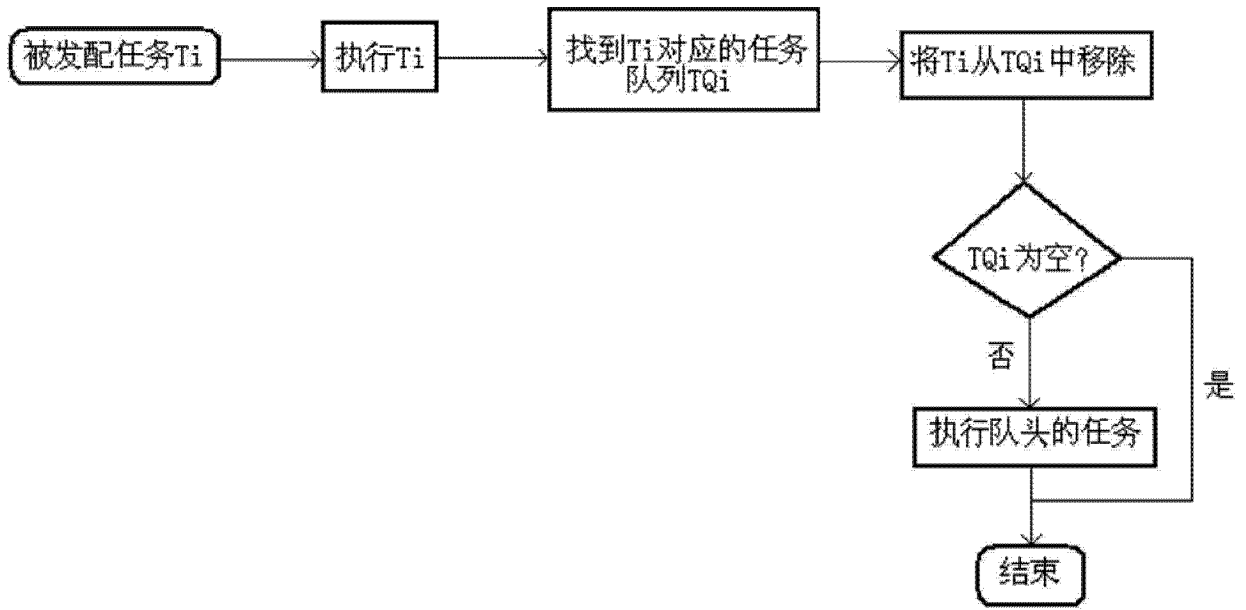


图 3