



(12) 发明专利

(10) 授权公告号 CN 101222510 B

(45) 授权公告日 2010. 11. 03

(21) 申请号 200810056824. 5

(22) 申请日 2008. 01. 25

(73) 专利权人 北京工业大学

地址 100022 北京市朝阳区平乐园 100 号

(72) 发明人 徐喆 闫士珍 宋威 余春暄

段建民 张明杰

(74) 专利代理机构 北京思海天达知识产权代理

有限公司 11203

代理人 刘萍

(51) Int. Cl.

H04L 29/06 (2006. 01)

H04L 12/28 (2006. 01)

(56) 对比文件

CN 1913476 A, 2007. 02. 14, 全文.

EP 1184754 A2, 2002. 03. 06, 全文.

US 2003/0084216 A1, 2003. 05. 01, 全文.

CN 1889567 A, 全文.

CN 1562654 A, 2005. 01. 12, 全文.

WO 01/20854 A1, 2001. 03. 22, 全文.

孙松, 孙川. 测控网络中 CAN 总线及其实现. 计量与测试技术 7. 2004, (7), 23-25.

审查员 王澍

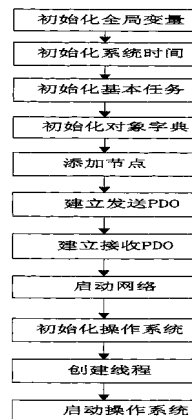
权利要求书 2 页 说明书 5 页 附图 5 页

(54) 发明名称

一种实现 CANopen 主站的方法

(57) 摘要

一种实现 CANopen 主站的方法属于现场总线领域。其特征在于本主站的实现是基于 MC9S12DP512 硬件平台, 在 UC/OS-II 操作系统之上构建的一个多线程系统, 包括调度机线程、写报文线程和读报文中断。调度机用来管理 CAN 报文接收、CAN 报文发送、PDO 报文分发、SDO 报文分发、NMT 报文分发、同步报文生成、PDO 处理、SDO 处理和 NMT 处理等 CANopen 的基本任务。另外, 作为 CANopen 协议核心的对象字典是通过散列表的方式构建的。这种构建对象字典的方式满足了主站对其对象字典必须具有可快速查找、添加、删除对象的特性的要求。本发明的优越性在于将任务调度应用在主站的设计中, 使其具有高度的实时性、并发处理能力和动态灵活性; 所实现的对象字典能满足网络的灵活扩展能力, 可动态的向对象字典中添加对象。



1. 一种实现 CANopen 主站的方法,其特征在于,包括以下步骤:

1) 主站开始运行时,首先初始化全局变量,对本协议栈中需要的所有全局变量进行初始化;

2) 初始化系统时间,该系统时间是通过定时器获得的微秒级时间;

3) 初始化基本任务,包括给它们分配存储空间,赋初始值;

4) 初始化对象字典,建立基本的对象字典项;

5) 添加节点,向主站的对象字典中添加从节点的信息;

6) 建立发送 PDO,在对象字典中添加发送 PDO 对应的所有信息;

7) 建立接收 PDO,在对象字典中添加接收 PDO 的所有信息;

8) 启动网络;

9) 初始化操作系统;

10) 创建线程,这里只创建一个开始线程;

11) 启动操作系统,操作系统开始调度线程:第一个被操作系统调度的线程是开始线程,在这个线程中完成的工作包括硬件的初始化、调度机线程的创建和写报文线程的创建,开始线程运行一次之后就将自己销毁;之后写报文线程被调度运行,但是此时没有报文要发送,所以写报文线程退出运行状态并触发调度机线程;调度机线程完成 CANopen 内部各任务的调度;在没有更高优先级线程处于就绪态或者出现中断时调度机线程将会一直处于循环运行状态;

当调度机中的任务需要发送报文时触发写报文发送线程,写报文发送线程将需要发送的报文写进 CAN 控制器相应的寄存器中并将其发送到 CAN 总线上;当 CAN 控制器收到报文时将产生报文接收中断,读报文中断服务程序开始执行,将报文从 CAN 控制器中读出,当调度机线程再次获运行时将调度其管理的相应任务处理该报文。

2. 根据权利要求 1 所述的一种实现 CANopen 主站的方法,其特征在于,采用散列表的方式构建对象字典:

使用一个数组来存储对象,设计一个哈希函数,使得每个对象的索引和子索引都与数组的一个下标相对应;数组定义如下:

$ODIndex\ iOD[HS\_index\_range]$

其中,HS\_index\_range 为散列地址的范围常量;

对象的索引和子索引通过哈希函数转化为散列地址;采用如下哈希函数:

$HS\_index = \text{mod}(((index * ran\_num) \gg shift\_num + sub\_index), HS\_index\_range)$

式中,HS\_index 为散列地址,mod() 为取模运算,index 为对象索引,ran\_num 为随即因子,shift\_num 为位移因子,sub\_index 为对象子索引,HS\_index\_range 为散列地址的范围常量;

当出现对于不同的对象,却计算出了相同的函数值,这样就产生了冲突,即从 index 和 sub\_index 到 HS\_index 的映射会重叠;采用链地址法处理冲突,将映射到同一个散列地址处多于一个的对象组成链表,即形成该散列地址对应的溢出表。

3. 根据权利要求 1 所述的一种实现 CANopen 主站的方法,其特征在于,调度机线程具体为:调度机通过任务等待队列和任务运行队列对 CANopen 的任务进行管理,有报文到时,CAN 报文接收任务运行,将接收到的报文放入 PDO、SDO、NMT 报文接收队列,并触发 PDO 报文

分发或 SDO 报文分发或 NMT 报文分发任务 ;PDO 报文分发、SDO 报文分发、NMT 报文分发任务将触发它们对应的处理任务 :PDO 处理、SDO 处理和 NMT 处理任务 ;如果在 PDO 处理、SDO 处理和 NMT 处理任务中有报文要发送或者同步报文生成任务正在运行执行,将会触发 CAN 报文发送任务。

## 一种实现 CANopen 主站的方法

### 技术领域

[0001] 本发明属于现场总线领域,特别是一种实现 CANopen 主站的方法。

### 背景技术

[0002] 作为一种真正开放的 CAN 总线高层协议,CANopen 协议允许不同的 CAN 设备以标准化的方式进行通讯,使得 CAN 设备具有互操作性。随着 CANopen 协议的日益完善,它在国内外已经有很多方面的应用,受到了广泛的重视。CANopen 目前正在从汽车工业向更多领域辐射,已经广泛应用到公共交通运输系统、医疗设备、海运电子设备和建筑自动化系统中。在欧洲,CANopen 协议已被广泛的应用于医疗装置中,并进一步扩展应用到保安控制系统中;在美国,CANopen 协议已经成为装载机械和公共运输设备的协议标准,同时也应用于嵌入式系统的控制。但是据了解,国内在 CANopen 的研究还处于起步阶段,除了中国单片机公共实验室 (BOL) 外,目前鲜有研究院所在进行相关工作,而中国在该领域的标准制定工作也仍处于摸索阶段。因此 CANopen 协议在我国的进一步推广仍需大量的研发工作。CANopen 主站的实现方法在国外也无相关技术的公开文献报道。

### 发明内容

[0003] 本发明的目的在于提供一种实现 CANopen 主站的方法。

[0004] 本发明提供一种实现 CANopen 主站的方法,如图 1 和图 2 所示,其特征在于,包括以下步骤:

[0005] 1) 主站开始运行时,首先初始化全局变量,对本协议栈中需要的所有全局变量进行初始化;

[0006] 2) 初始化系统时间,该系统时间是通过定时器获得的微秒级时间;

[0007] 3) 初始化基本任务,包括给它们分配存储空间,赋初始值;

[0008] 4) 初始化对象字典,建立基本的对象字典项;

[0009] 5) 添加节点,向主站的对象字典中添加从节点的信息;

[0010] 6) 建立发送 PDO,在对象字典中添加发送 PDO 对应的所有信息;

[0011] 7) 建立接收 PDO,在对象字典中添加接收 PDO 的所有信息;

[0012] 8) 启动网络;

[0013] 9) 初始化操作系统;

[0014] 10) 创建线程,这里只创建一个开始线程;

[0015] 11) 启动操作系统,操作系统开始调度线程:第一个被操作系统调度的线程是开始线程,在这个线程中完成的工作包括硬件的初始化、调度机线程的创建和写报文线程的创建,该线程运行一次之后就将自己销毁。之后写报文线程被调度运行,但是此时没有报文要发送,所以该线程退出运行状态并触发调度机线程。调度机线程完成 CANopen 内部各任务的调度。在没有更高优先级线程处于就绪态或者出现中断时该线程将会一直处于循环运行状态。

[0016] 当调度机中的任务需要发送报文时触发写报文发送线程,该线程将需要发送的报文写进 CAN 控制器相应的寄存器中并将其发送到 CAN 总线上。当 CAN 控制器收到报文时将产生报文接收中断,读报文中断服务程序开始执行,将报文从 CAN 控制器中读出,当调度机线程再次获运行时将调度其管理的相应任务处理该报文。

[0017] 对象字典是 CANopen 协议的核心,是一个有序的对象组,其中定义了 CANopen 网络中设备的所有信息。在 CANopen 网络中主站要行使网络管理、配置从站的功能。主站对象字典的构建必须考虑网络的灵活扩展能力,即,灵活的增减节点。网络中节点的增减必然要求修改主站对象字典,甚至大量添加、删除对象字典中的对象。所以主站的对象字典必须具有可快速查找、添加、删除对象的特性。

[0018] 本发明采用散列表的方式构建对象字典。具有可动态更新、搜索效率高和存储空间利用率高等优点。

[0019] 具体采用散列表的方式构建对象字典如下:

[0020] 使用一个数组来存储对象,设计一个哈希函数,使得每个对象的索引和子索引都与数组的一个下标相对应;数组定义如下:

[0021]  $ODIndex\ iOD[HS\_index\_range]$

[0022] 其中,  $HS\_index\_range$  为散列地址的范围常量;

[0023] 对象的索引和子索引通过哈希函数转化为散列地址。采用如下哈希函数:

[0024]  $HS\_index = \text{mod}(((index * ran\_num) \gg \gg shift\_num + sub\_index), HS\_index\_range)$

[0025] 式中,  $HS\_index$  为散列地址,  $\text{mod}()$  为取模运算,  $index$  为对象索引,  $ran\_num$  为随即因子,  $shift\_num$  为位移因子,  $sub\_index$  为对象子索引,  $HS\_index\_range$  为散列地址的范围常量。

[0026] 当出现对于不同的对象,却计算出了相同的函数值,这样就产生了冲突,即从  $index$  和  $sub\_index$  到  $HS\_index$  的映射会重叠;采用链地址法处理冲突,将映射到同一个散列地址处多于一个的对象组成链表,即形成该散列地址对应的溢出表。

[0027] 为了方便实现 CANopen 协议,本主站按功能定义了 CAN 报文接收、CAN 报文发送、PDO 报文分发、SDO 报文分发、NMT 报文分发、同步报文生成、PDO 处理、SDO 处理和 NMT 处理等任务。根据主站的实时性和可扩展性的要求,采用多任务调度的核心结构,即,调度机通过任务等待队列和任务运行队列对 CANopen 的任务进行管理,如图 3 所示,有报文到时,CAN 报文接收任务运行,将接收到的报文放入 PDO、SDO、NMT 报文接收队列,并触发 PDO 报文分发或 SDO 报文分发或 NMT 报文分发任务。PDO 报文分发、SDO 报文分发、NMT 报文分发任务将触发它们对应的处理任务:PDO 处理、SDO 处理和 NMT 处理任务。如果在 PDO 处理、SDO 处理和 NMT 处理任务中有报文要发送或者同步报文生成任务正在运行执行,将会触发 CAN 报文发送任务。

[0028] 调度机线程具体为:调度机通过任务等待队列和任务运行队列对 CANopen 的任务进行管理,有报文到时,CAN 报文接收任务运行,将接收到的报文放入 PDO、SDO、NMT 报文接收队列,并触发 PDO 报文分发或 SDO 报文分发或 NMT 报文分发任务。PDO 报文分发、SDO 报文分发、NMT 报文分发任务将触发它们对应的处理任务:PDO 处理、SDO 处理和 NMT 处理任务。如果在 PDO 处理、SDO 处理和 NMT 处理任务中有报文要发送或者同步报文生成任务正

在运行执行,将会触发 CAN 报文发送任务。

### 附图说明

- [0029] 图 1CANopen 主站系统结构图
- [0030] 图 2 系统软件流程图
- [0031] 图 3 调度机内部任务之间关系结构图
- [0032] 图 4 调度机结构图
- [0033] 图 5 任务运行流程图
- [0034] 图 6 对象字典逻辑结构图
- [0035] 图 7 随机因子、位移因子和搜索时间之间的关系
- [0036] 图 8 引入高速缓存思想前后搜索效率的比较

### 具体实施方式

[0037] 本发明是基于飞思卡尔 16 位单片机 MC9S12DP512 硬件平台的 CANopen 主站的开发,系统软件结构框图如图 1 所示。其中开始线程、调度机线程、写报文线程是用户建立的,由 UC/OS-II 内核管理的三个线程。开始线程是操作系统完成初始化之后所调度的第一个线程。它主要完成硬件的初始化,包括串口,系统时钟的初始化、时钟节拍的初始化、CAN 的初始化、中断的初始化以及创建调度机线程、写报文线程。该线程运行一次之后就将自己销毁。调度机线程主要完成 CANopen 内部各任务的调度。在没有更高级线程处于就绪态或者出现中断时该线程将会一直处于循环运行状态。写报文线程负责将需要发送的报文写进 CAN 控制器相应的寄存器中并将其发送到 CAN 总线上。该线程由调度机线程在需要的时候启动。

[0038] 调度机管理的任务包括 :CAN 报文接收、CAN 报文发送、PDO 报文分发、SDO 报文分发、NMT 报文分发、同步报文生成、PDO 处理、SDO 处理和 NMT 处理等任务。调度机通过任务等待队列和任务运行队列对它们进行管理。它们之间的调度关系如图 2 所示 :

[0039] 有报文到时,CAN 报文接收任务运行,将接收到的报文放入 PDO、SDO、NMT 报文接收队列,并触发 PDO 报文分发或 SDO 报文分发或 NMT 报文分发任务。PDO 报文分发、SDO 报文分发、NMT 报文分发任务将触发它们对应的处理任务 :PDO 处理、SDO 处理和 NMT 处理任务。如果在 PDO 处理、SDO 处理和 NMT 处理任务中有报文要发送或者同步报文生成任务正在运行执行,将会触发 CAN 报文发送任务。

[0040] 调度机的结构如图 4 所示。该调度机有两个基本的队列组成 :任务等待队列、任务运行队列。其中,任务运行队列设计采用和 UNIX SVR4 相类似的运行队列结构。一共有九个优先级,则同时存在 9 个优先级队列,每一个队列有一个标志位。当队列中包含可运行任务,标志位为 1,否则为 0。其中,执行选择、执行任务、销毁任务、运行选择由调度机主函数完成,构成一个循环体。

[0041] 对于单个任务而言,其运行的流程如图 5 所示。一旦任务获得执行,它的运行、停止、终结都由任务自身决定。一旦开始执行,主函数并不会主动终止函数。通过不同的返回信息,主函数可以知道如何对任务进行处理。本系统规定任务有以下几种返回值 :

[0042] TASK\_OK :任务运行完毕,可以销毁 ;

[0043] TASK\_WAIT :任务自身阻塞,放入等待队列;

[0044] TASK\_RUN :任务可继续运行,放入执行队列;

[0045] TASK\_ERR :任务运行出错,进行异常处理。

[0046] 本发明采用散列表的方式构建对象字典,使用一个下标范围比较大的数组来存储对象。可以设计一个函数(哈希函数,也叫做散列函数),使得每个对象的索引和子索引都与一个函数值(即数组下标)相对应,于是用这个数组单元来存储这个对象;也可以简单的理解为,按照索引和子索引为每一个对象“分类”,然后将这个对象存储在相应“类”所对应的地方。数组定义如下:

[0047] ODIndex iOD[HT\_WIDTH]

[0048] 其中,HT\_WIDTH 为散列地址的范围常量。

[0049] 对象的索引和子索引通过哈希函数转化为散列地址。哈希函数应尽可能的使索引和子索引经过哈希函数得到一个随机的地址,以便将对象均匀的分布到整个地址区间中;由于对象字典访问的频繁性,映射函数还要足够简单以减少运算时间,这里采用如下哈希函数:

[0050]  $HS\_index = \text{mod}(((index * ran\_num) \gg \gg shift\_num + sub\_index), HS\_index\_range)$  式 1

[0051] 式 1 中,HS\_index 为散列地址,mod() 为取模运算,index 为对象索引,ran\_num 为随即因子,shift\_num 为位移因子,sub\_index 为对象子索引,HS\_index\_range 为散列地址的范围常量。

[0052] 不能够保证每个对象的索引和子索引与函数值是一一对应的,因此极有可能出现对于不同的对象,却计算出了相同的函数值,这样就产生了冲突,即从 index 和 sub\_index 到 HS\_index 的映射会重叠。这里处理冲突的方法采用链地址法,将映射到同一个散列地址处多于一个的对象组成链表,即形成该散列地址对应的溢出表。从散列表的相应位置可以访问其对应的溢出表。

[0053] 用散列表构建的对象字典逻辑结构图如图 6 所示,散列表地址范围为 N, E 代表一个索引项, N 代表一个未使用的散列表项。

[0054] 根据统计,一个小型的 CANopen 网络, CANopen 主站需要保存约 1500 个数据字典项。这里选择散列表空间大小为 512,即 HS\_index\_range 取值为 512,这样,平均每个散列表项只需要链接一个长度为 2 的溢出表。在没有引入大的搜索时间的情况下,合理利用了空间。

[0055] 可以根据现有的初始对象字典进行仿真,如图 7 所示,其中搜索时间的计算公式如下:

[0056]  $sum = \sum_{i=1}^{512} \sum_{k=0}^{n(i)} k$  式 2

[0057] 式 2 中 sum 为遍历所有初始化对象需要的搜索时间, n(i) 代表样本空间中映射到散列地址 i 的对象个数。从图中可以得到一个搜索时间最小点,即当 ran\_num 取 193、shift\_num 取 3 时搜索时间最小,散列表填充率最高,可达 99.8%。

[0058] 为了进一步提高搜索效率,借鉴高速缓存的思想,如果访问溢出表中某一结点需要的比较次数大于阈值 OD\_TARGET\_DEPTH,则将该结点移到溢出表表头的位置。从而可以将

经常被访问的对象存放在溢出表中靠前的位置。当再次访问这些对象时,可以快速从溢出表获得,从而进一步提高了搜索效率。此处 OD\_TARGET\_DEPTH 的取值应根据所构建的对象字典中常用的对象个数取值,应满足溢出表的前 OD\_TARGET\_DEPTH 项能够容纳所有常用的对象。

[0059] 引入高速缓存的思想前后搜索效率比较如图 8 所示。图中横坐标表示对象的个数,纵坐标表示搜索常用对象(随机取总的对象个数的 1/4 作为常用对象)的平均比较次数。其中曲线 1 为没有引入高速缓存思想时所得到的曲线;曲线 2 表示引入了高速缓存思想后得到的曲线。显然,引入高速缓存思想后搜索效率得到了明显提高。



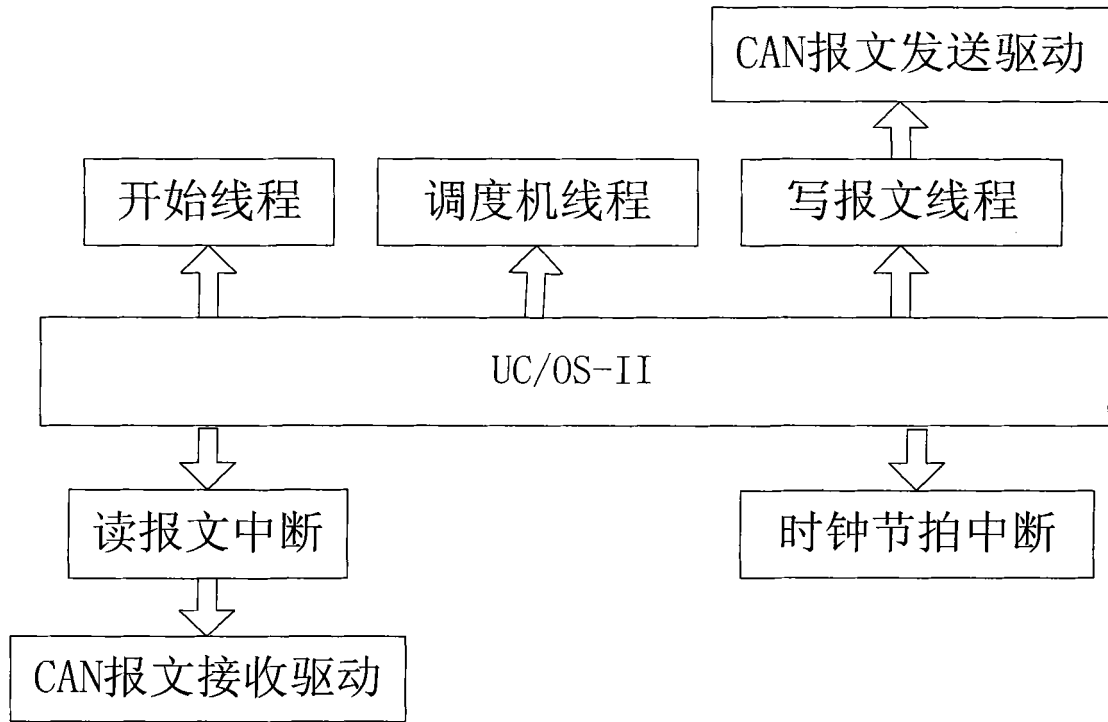


图 1



图 2

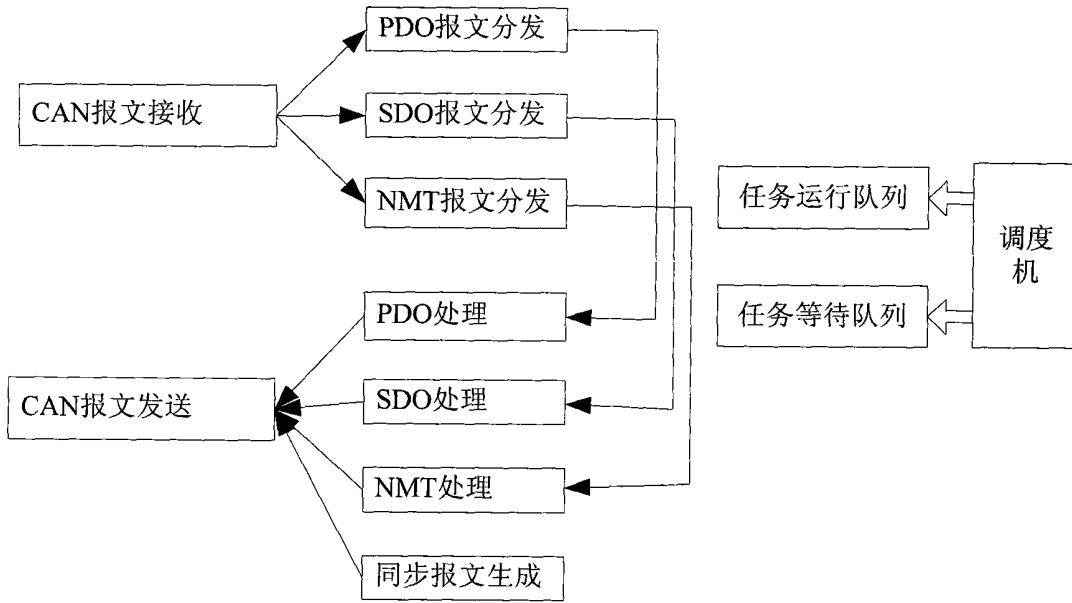


图 3

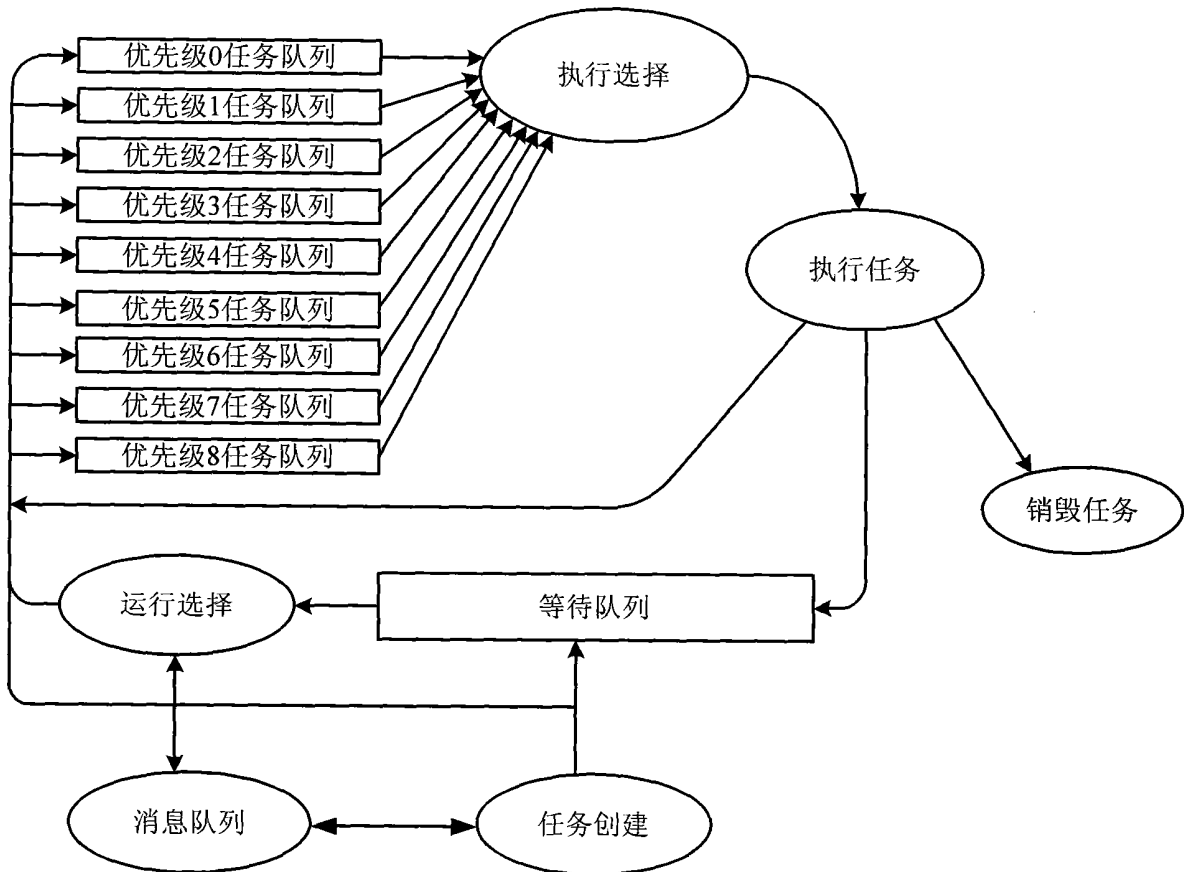


图 4

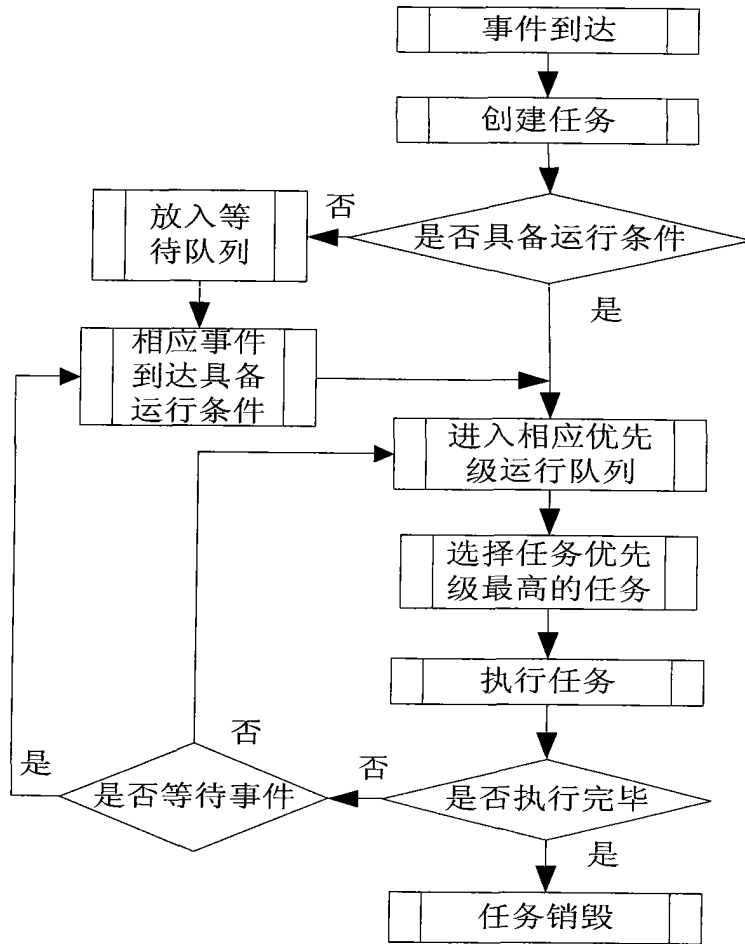


图 5

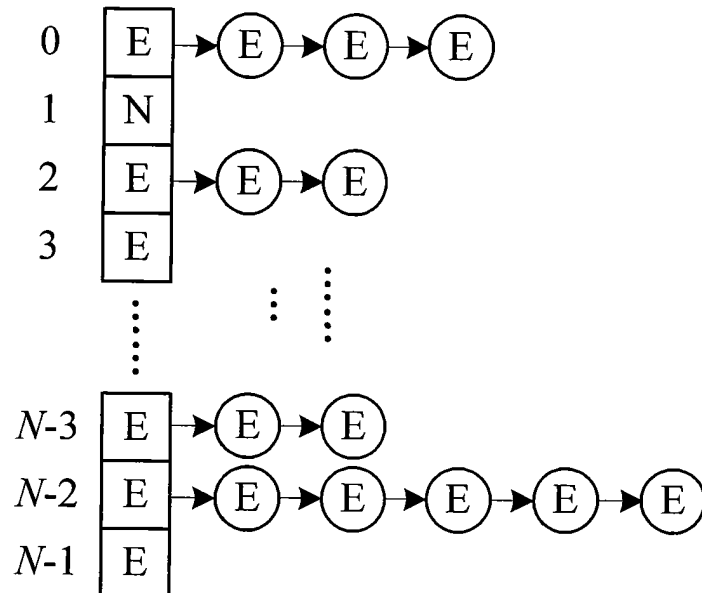


图 6

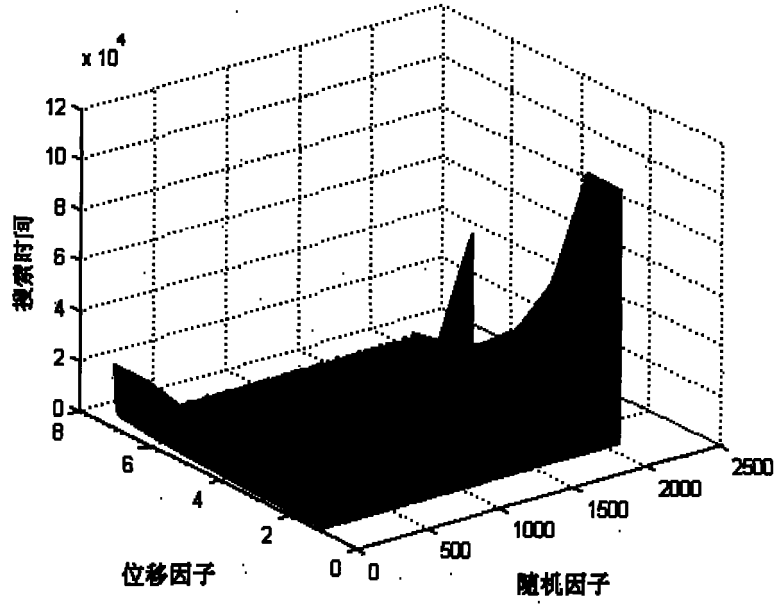


图 7

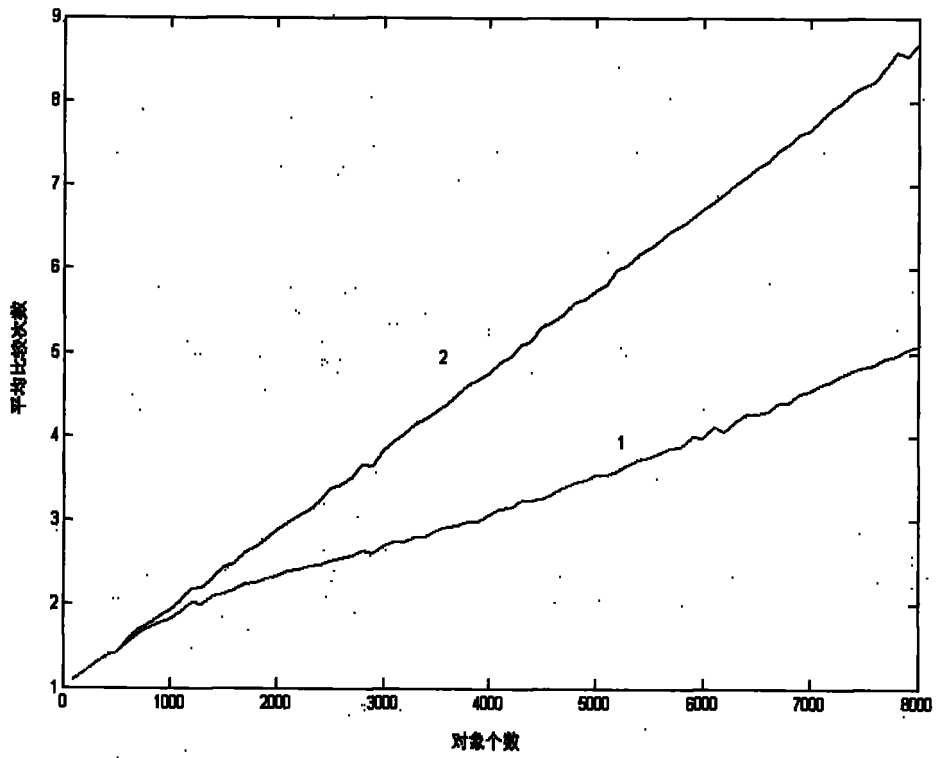


图 8