

一种新型的实时调度算法

宋杰, 檀林欣, 曹竹冬, 王书菊

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039)

摘要:在嵌入式系统中,系统的实时性很大程度上取决于任务调度策略。对如何提高系统实时性的研究已经成为该领域的一大热点。文中针对传统的忽略了作业所能承受的等待时间的能力仅以等待时间的长短而来衡量优先级的先来先服务(FCFS)算法,和仅以任务的周期长短来设置优先级而导致那些周期长但急需执行的任务无法被调度的速率单调(RM)算法的缺陷,提出一种新的动态设置优先级的方法—剩余时间法。该方法以任务的等待时间和周期共同来决定优先级。通过分析和实验表明该方法兼具RM和FCFS的优点。它不仅考虑到短周期任务应优先执行,而且同时也把任务等待时间作为评价优先级的重要因素,实验结果证实了该方法能更有效地提高实时任务的完成率和CPU的利用率。

关键词:等待时间;周期;RM算法;剩余时间算法;任务的完成率;CPU利用率

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2010)12-0073-04

A New Real-Time Scheduling Algorithm

SONG Jie, TAN Lin-xin, CAO Zhu-dong, WANG Shu-ju

(Ministry of Edu. Key Lab. of Intelligent Computing & Signal Processing, Anhui Univ., Hefei 230039, China)

Abstract: In embedded operation system, system's performance largely depends on the scheduling strategy. The research on how to improve the system's performance to satisfy the real-time requirement has become a hotspot in this area. For the traditional first come first server(FCFS) scheduling algorithm determines priority only by the length of waiting-time but ignores the endurance of the job's waiting-time, and rate monotonic(RM) algorithm just measure jobs' priority by the length of jobs' life cycle but makes those who have a long life period and urgently needed to be executed can't be scheduled, aiming at those, a new method of setting priorities dynamically—the rest time algorithm is proposed in this paper. This algorithm determines priority by jobs' waiting-time and life cycle together. Through analysis and experiments see that this method possesses the advantage of both RM and FCFS. In this method, tasks are given priority not only according to the length of life-cycle, but also considering the task's waiting-time. Experimental results show that it can largely improve the real-time task's completion rate and CPU's utilization.

Key words: waiting-time; life-cycle; RM algorithm; rest time algorithm; task's completion rate; CPU's utilization

0 引言

随着嵌入式系统的不断发展和广泛应用,对其提出的实时性^[1,2]要求也越来越高。嵌入式实时操作系统既是一种嵌入式操作系统又是一种实时操作系统。从嵌入式操作系统角度而言,它的特点是:低功耗和可裁剪;而作为实时操作系统而言,更重要的是要能及时地处理和响应事务,从而达到实时性的要求。所谓的实时性是指系统能够在限定的时间内完成任务并对外部的异步事件做出及时响应。对于如何在规定的时间内,尽可能更快地完成给定的任务,以满足实时性的要求,一直是一个重要的研究课题。而任务调度策

略在很大程度上决定了系统实时性的好坏。近年来,任务调度方面的研究成果层出不穷,产生了许多实时调度算法^[3-6]。

在实时任务调度中,可以用任务的等待时间来做为优先级的评价标准,也就是说任务等待的时间越长,它的优先级越高。例如FCFS算法^[7],虽然这类算法容易实现,但效率不高,只顾及作业等候时间,没考虑作业所能承受等待时间的长短。另外也可以根据任务的执行周期的长短来决定调度优先级,那些具有较短的执行周期的任务具有较高的优先级。例如RM算法^[1,8,9,10,11],RM算法具有运行开销小和能将频率高的任务顺利完成的特点,但没有考虑到任务的等待时间,造成那些周期长但急需执行的任务无法被调度。基于这两种算法的优缺点,文中提出了一种新的算法,它用任务的等待时间和周期共同来决定优先级。具体

收稿日期:2010-04-09;修回日期:2010-07-13

基金项目:安徽省教育科研项目(2008jyxm277)

作者简介:宋杰(1966-),男,副教授,硕士生导师,研究方向为嵌入式系统、计算机原理与接口、生物信息学。

地说是以周期减去等待时间的差值来衡量优先级,在文中称这个差值为剩余时间,剩余时间越小,优先级越高。实验证明剩余时间法能够很好地提高任务调度的成功率和 CPU 的利用率。

1 RM 算法

RM 算法于 1973 年是由 Liu 和 J. Layland 提出的^[12],RM 算法的核心思想是按照任务的周期长短为每个任务分配一个固定的优先级,分配原则是:周期越短,优先级越高。

为了简化,RM 算法对任务做了如下的三点假设^[1~9]:

- (1)所有任务都是周期性的,并且周期是已知的;
- (2)任务间没有数据交换、资源共享等问题,各任务之间相互独立;
- (3)CPU 采用可抢占式调度算法,即优先执行优先级最高的就绪状态的任务。

下面的公式(1)被 Liu 等人证明是 RM 算法可行性的充分条件^[4]:对于任务集 $S = (r_1, r_2, \dots, r_n)$,若这个任务集的 CPU 利用率满足下面公式(1),那么这个任务集用 RM 调度是可行的,即可以用 RM 算法进行调度。

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_n}{T_n} \leq n(2^{1/n} - 1) = L(n) \quad (1)$$

公式(1)中 C_n 为任务 r_n 的执行时间, T_n 为任务 r_n 的周期。 $L(n)$ 为 n 个任务的 CPU 利用率的最小上界。

RM 算法具有可调度性测试简单和系统开销小等优点。但没有考虑到任务的等待时间,造成那些周期长但急需执行的任务无法被调度。

2 剩余时间算法

2.1 算法描述

在该算法中,任务的剩余时间(T_s)定义为任务的周期(T)减去该任务的等待时间(T_w),即 $T_s = T - T_w$ 。该算法的思想是:在每一次任务调度前,计算每一个等待任务的剩余时间,并按照各任务的剩余时间长短为每个任务分配或重新分配一个优先级,分配原则是:剩余时间越短,优先级越高。其中任务的等待时间是动态变化的。

该算法分几种情况讨论:

- 1)按任务的剩余时间大小对任务的优先级进行设置,调度优先级最高即剩余时间最短的任务执行;
- 2)如果当前优先级队列中优先级最高的任务的剩余时间大于该任务需要运行的时间,则该任务被调用执行;否则不予调度执行;

3)如果几个任务的剩余时间相等,也就是说它们的优先级相同,则按照任务需要运行时间的大小来决定调用任务的先后顺序。此时调用需要运行的时间最小的任务来执行,然后调用需要运行的时间次小的任务,以此类推。这样可以保证更多的任务被执行,能更有效地提高 CPU 利用率;

4)如果几个任务的剩余时间相等,且它们需要运行的时间也相等,那么就按照先来先服务的原则进行任务调度。

2.2 算法流程

文中假设某个时间基的滴答数(tick)为时间单位。

- (1)初始化各任务的任务控制块和初始时间;
- (2)初始时所有任务的等待时间为零,根据文中剩余时间的定义,此时任务的剩余时间就等于任务自身的周期,依据任务的剩余时间长短来设置任务的优先级,这时候的调度算法实际上变成了 RM 算法;
- (3)更新计数器;
- (4)调度优先级最高的任务执行,判断该任务的剩余时间和它执行时需要运行的时间,如果剩余时间小于执行时间,则该任务不被执行。将其暂时从优先队列里删除。继续从优先队列里调度下一个任务来进行相同的判断。如果找到了满足条件的任务,假设这个任务为 p_1 ,则执行 p_1 ,待 p_1 执行完后,修改其他未执行的任务的等待时间,等待时间 = 原来等待时间 + p_1 的执行时间。如果这个时候有新的任务到来,将新任务的等待时间赋值为零;

(5)重复步骤(3),再次调度。

为了清晰起见,整个算法的流程图如图 1 所示。

2.3 算法的假设条件

为了简化分析,对任务集作如下几点假设:

- (1)所有的任务是周期性的,并且具有已知的周期;
- (2)所有任务的相对截止期限等于任务的周期;
- (3)任何一个任务的执行不依赖于其他任务请求的开始或者结束,各任务之间相互独立;
- (4)所有任务的运行时间是固定不变的,这里假定任务是在 CPU 无中断情况下运行的;
- (5)忽略所有系统的开销时间(包括上下文切换在内和中断处理的时间开销);
- (6)所有任务的调度都是在单处理器上进行。

3 算法的性能测试及实验

为了测试文中提出的剩余时间算法的性能,将通过实验对 RM 算法和剩余时间算法的性能进行比较来验证剩余时间算法的实时调度性能。

假设任务 P_i 的周期为 T_i , 在一个给定的时间段 t 内, P_i 的理想执行次数应该是 $\lfloor t/T_i \rfloor$, 而实际上系统内很有可能不止 P_i 这一个任务在执行, 通过剩余时间算法调度不同的任务来执行, 所有任务的优先级都在不停地动态改变, 那么就存在任务 P_i 在它的相对截止期限内不被系统调度执行的可能性。也就是说 P_i 实际执行的次数可能比理想执行次数少。把实时任务实际被调度执行的次数记为 $N_{(实际)}$, 把任务的理想执行次数记为 $N_{(理想)}$, 将 $N_{(实际)}/N_{(理想)}$ 的比值称为任务在截止期限内被满足的程度, 也就是任务完成率^[1], 它是衡量实时调度算法的一个重要指标, 它的值为 0 ~ 100%。

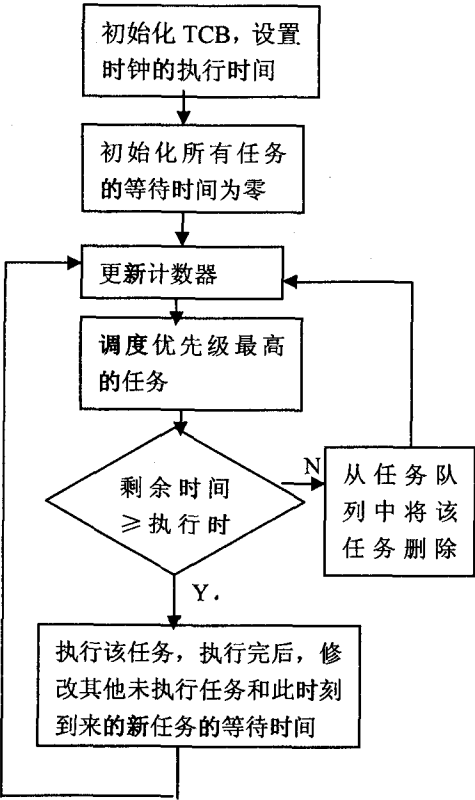


图 1 算法流程图

在实验中, 实时任务的完成率及 CPU 的利用率是衡量剩余时间算法的重点对象。因此, 在本实验中, 主要对比任务集在 RM 算法和剩余时间算法下的各任务完成率和 CPU 的利用率。

本实验选择在单处理器上进行, 这样便于更清楚地观察任务的运行情况。设有两个任务集 $S = \{S1, S2, S3\}$ 和 $T = \{T1, T2, T3\}$, 任务集 S 和 T 中各任务的周期和运行时间分别如下面的表 1 和表 2 所示。实验数据集来源为董吉文, 张阳的“嵌入式实时操作系统任务调度算法的改进与应用”一文中的实验数据, 见参考文献[1]。总运行时间为 8400 个时间单位, 最终的实验结果如表 3 和表 4 所示。

表 1 任务集 S

任务集 S	周期	运行时间
S1	80	5
S2	50	10
S3	60	40

表 2 任务集 T

任务集 T	周期	运行时间
T1	60	15
T2	20	10
T3	40	10

表 3 任务集 S 在两种算法下的实验结果比较

任务集 S	RM 算法的任务完成率	剩余时间算法的任务完成率
S1	86.67%	100%
S2	100%	100%
S3	100%	100%
CPU 利用率	92.08%	92.9167%

表 4 任务集 T 在两种算法下的实验结果比较

任务集 T	RM 算法的任务完成率	剩余时间算法的任务完成率
T1	50%	100%
T2	100%	100%
T3	100%	100%
CPU 利用率	87.5%	100%

从上面的实验可以看出, RM 算法对短周期的任务能很好地保证其完成率, 但对于长周期的任务则显得力不从心。而文中提出的剩余时间算法, 在确定任务调度优先级时, 不仅考虑到短周期任务应优先执行, 而且同时也把任务等待时间作为评价优先级的重要因素。

通过表 3、表 4, 从算法对任务的完成率上, 可以看出, RM 算法对任务集中周期长的任务(如任务集 S 中的 $S1$, 任务集 T 中的 $T1$) 的调度, 很不理想, $S1$ 和 $T1$ 的任务完成率分别为 86.67% 和 50%。而文中所提出的算法对 $S1$ 和 $T1$ 的任务完成率都达到了 100%。从算法对 CPU 的利用率上, 可以看出, RM 算法对 CPU 的利用率分别为 92.08% 和 87.5%。而文中算法对 CPU 的利用率分别为 92.9167% 和 100%。显然文中算法在任务完成率和 CPU 利用率上有较大的提高, 能较大地减少 CPU 的空跑时间和更充分地利用资源。

4 结束语

RM 算法和 FCFS 算法是较为经典的任务调度算法。RM 作为一种静态优先级调度算法, 在运行前确

定任务的执行顺序,实现简单,在其可调度性界限内,可以保证任务的成功调度。具有运行开销小和能将频率高的任务顺利完成的特点。但是,RM 调度算法没有考虑到任务的等待时间,造成那些周期长但急需要执行的任务无法被调度。FCFS 算法仅用任务的等待时间来作为优先级的评价标准,这类算法容易实现,但效率不高,只顾及作业等候时间,没考虑作业所能承受等待时间的长短。文中针对这几种传统的只以任务的等待时间或者只以任务的周期来设置优先级,提出了一种根据任务的等待时间和周期来共同决定优先级的算法—剩余时间法,该算法根据任务的周期与等待时间的差值来动态修改任务的优先级。这样做既能考虑到任务的等待时间,也能考虑到任务所能容忍的等待时间的极限。并且在文章的后半部分,通过任务调度实验与 RM 算法进行比较,实验结果证明剩余时间算法能很好地提高任务的完成率及 CPU 的利用率。

参考文献:

- [1] 董吉文,张 阳.嵌入式实时操作系统任务调度算法的改进与应用[J].计算机应用,2009(9):2516-2519.
- [2] 陈文星,张辉宣,陶 陶,等.嵌入式 Linux 的实时性改进技术[J].计算机技术与发展,2006,16(10):114-117.

(上接第 72 页)

智能化低的弱点,同时通过利用 WordNet 减轻了在语义领域中对于每一领域都需建立本体库的复杂工作。下一阶段工作将尝试引入 WordNet 概念链中除了上下位关系以外的多种关系,并在服务匹配过程中综合考虑服务的多种属性。

参考文献:

- [1] 白东伟.基于语义的 web 服务发现与技术研究[D].北京:北京邮电大学,2007.
- [2] 胡建强,邹 鹏,王怀民,等. Web 服务描述语言 QWSDL 和服务匹配模型研究[J].计算机学报,2005,28(4): 505-513.
- [3] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval[M]. [s. l.]: Addison Wesley, 1999.
- [4] Stroulia E, Wang Y. Structural and semantic matching for assessing Web service similarity[J]. International journal of cooperative information systems, 2005, 14(4): 407-438.
- [5] Kokash N A comparison of Web service Interface similarity measures[C]//3rd European starting AI researcher symposium. Riva delGarda, Italy: IOS Press, 2006: 220-231.
- [6] 张丽坤,蒋 波.基于本体的语义 Web 研究[J].计算机技

- [3] 刘胜辉,马 嵩.基于 Linux 内核的实时调度机制研究及应用[J].计算机工程与应用,2008,44(6):121-123.
- [4] 谢 敏,李桥梁.嵌入式实时操作系统任务调度算法优化[J].电子科技,2005(12):24-26.
- [5] Gafford J. Rate monotonic scheduling[J]. IEEE Micro, 1991, 11(3):34-39.
- [6] 邢科群,郝红卫,温天江.两种经典实时调度算法的研究与实现[J].计算机工程与设计,2006,27(1):117-119.
- [7] 赵俊锋.一种基于 FCFS 调度策略的多处理机系统的仿真模型[J].宁夏大学学报:自然科学版,2001(4):415-418.
- [8] 洪雪玉,张 凌,袁 华. Linux 下的实时调度算法[J].华南理工大学学报:自然科学版,2008,36(4):14-19.
- [9] 王永吉,陈秋萍.单调速率及其扩展算法的可调度性判定[J].软件学报,2004(6):799-814.
- [10] Obenza R. Rate monotonic analysis for real time systems[J]. IEEE Computer, 1993, 26(3):73-74.
- [11] Brandt S A, Banachowski S, Lin Caixue, et al. Dynamic integrated scheduling of hard real-time, soft real-time and non-real-time processes[C]//Proceedings of the 24th IEEE Real-Time Systems Symposium (RTSS 2003). [s. l.]: [s. n.], 2003.
- [12] Liu C L, Layland J. Scheduling algorithms for multiprogramming in a hard real-time environment[J]. J. ACM, 1973, 20(1):46-61.

术与发展,2007,17(6):116-119.

- [7] Paolucci M, Kawamura T, Payne T, et al. Semantic Matching of Web services Capabilities[C]//Proceedings of the International Semantic Web Conference. Sardinia, Italy: [s. n.], 2002: 333-347.
- [8] 陈江锋,于建军.基于扩展 VSM 的 Web 服务发现[J].计算机工程,2008,34(12):25-27.
- [9] Voorhees E. Using WordNet for Text Retrieval[C]//Fellbaum C. in WordNet: An Electronic Lexical Database 1998. The Cambridge, MA: MIT Press, 1999: 285-303.
- [10] 张 剑,李春平.基于 WordNet 概念向量空间模型的文本分类[J].计算机工程与应用,2006,42(4):174-179.
- [11] Wu Zhibiao, Palmer M. Verb semantics and lexical selection [C]//Inproceedings of the 32nd Annual Meeting of the Association for Computational Linguistics. New Mexico: [s. n.], 1994:133-138.
- [12] Vapnik V N. The Nature of Statistical Learning Theory[M]. New York: Springer-Verlag Inc, 1995.
- [13] W3C. Web Services Description Language, Version 1.1[EB/OL]. 2001. <http://www.w3.org/tr/WSDL>.
- [14] 叶 蕾,张 斌.基于功能语义的 web 服务发现方法[J].计算机研究与发展,2007,44(8):1357-1364.