

基于组合优先级的自适应实时调度算法研究

游文铃, 邵清

(上海理工大学 光电信息与计算机工程学院, 上海 200093)

摘要:在嵌入式实时操作系统中,由单个特征参数作为实时任务优先级的调度依据,并不能较好地描述系统中任务的关键性和紧迫性。提出一种基于组合优先级的自适应实时调度算法(SREDF),综合任务的截止期和 CPU 运行期设计任务的优先级,使截止期越早且 CPU 运行期越短的任务拥有最高优先级。处理器能有效地调度相同截止期的实时任务,并提前分析和预测任务能否完成。实验表明,该算法降低了任务调度的截止期错失率(MDP),并提高了任务调度的速度和效率。

关键词:实时操作系统;调度算法;任务优先级;截止期错失率

中图分类号:TP316 **文献标识码:**A **文章编号:**1000-8829(2015)04-0078-03

Self-Adaptive Real-Time Schedule Algorithm Based on Combinational Priority Strategy

YOU Wen-ling, SHAO Qing

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In the embedded real-time operating system, it is difficult to describe the importance and urgency of the task well by regarding single parameter as task priority of the schedule algorithm. A self-adaptive real-time schedule algorithm based on combinational priority strategy (SREDF) is proposed, which combines with the deadline and the runtime period, and sets the task which possesses the earliest deadline and the shortest of the CPU runtime the highest priority. The processor schedules those tasks with the same deadline, moreover, those tasks are analyzed and predicted in advance to judge whether they could be finished before the deadline. The experimental results show that the proposed algorithm decreases the missed deadline percentage(MDP) and improve the rate and efficiency of the task scheduling.

Key words: real-time operating system; schedule algorithm; task priority; missed deadline percentage(MDP)

在嵌入式实时操作系统中,任务的调度算法一直是研究的热点,大量的研究都集中在调度算法的应用。使用最广泛的调度策略是基于优先级的调度算法。基于优先级的调度算法主要包括单调速率算法(RM, rate-monotonic)^[1]、截止期最早最优先(EDF, earliest deadline first)^[2-4]、空闲时间最短最优先(LSF, least slack first)^[5-6]、最短 CPU 运行期优先、价值最高优先、价值密度最大最优先等策略。在上述的算法中,任

务的优先级都是基于一个或多个特征参数,如截止期、空闲时间或关键性等,然而,若优先级仅仅由某个特征参数来确定是不够的。

近年来,学者们提出了混合调度算法,如 EDV (earliest deadline value) 和 VED (value earliest deadline) 算法^[7]综合了价值和截止期为优先级依据的特征参数,采用两个特征参数作为调度算法的优先级,从而改进 EDF 的调度算法。文献[8]基于 DPA 的抢占调度算法(DRTP),根据任务的价值和剩余执行时间讨论剩余价值,再根据任务截止期和空余执行时间分析任务的紧迫性,综合考虑任务的剩余价值和执行的紧迫性。除此之外,学者们还提出了基于阈值的算法,如模糊阈值 EDF 调度算法^[9]通过延长当前任务截止期至动态模糊阈值来改进 EDF 算法。目前,许多的算法都仅仅考虑了系统的任务间的重要程度而忽略了系统完成任务的数量。

收稿日期:2014-04-10

基金项目:国家自然科学基金资助项目(61170277);上海市教委科研创新项目(12YZ094)

作者简介:游文铃(1986—),女,台湾人,硕士研究生,主要研究方向为实时调度算法、实时嵌入式操作系统;邵清(1970—),女,江苏人,博士,副教授,主要研究方向为嵌入式操作系统、容错计算。

为了改善上述算法存在的问题,笔者提出一种基于组合优先级的自适应实时调度算法(SREDF),将基本任务的截止期和 CPU 运行期这两个特征参数结合起来,综合设计任务的优先级,使得截止期越早且 CPU 运行期越短的任务拥有最高优先级。该算法适用于任务在任何时刻都可以被抢占的情况,并且任务间的结束时间相互独立,任务在抢占过程中对处理器的负载没有影响。实验表明,SREDF 算法的任务截止期错失率较之前的算法有明显下降,同时系统完成任务的数量和 CPU 的资源利用率得到了提高。

1 任务模型的建立

本文通过定义任务的特征、截止时间、剩余时间和所需时间,便于描述 SREDF 算法和实验仿真。

定义1 一个含有 n 个任务的集合为 $S = (T_1, T_2, \dots, T_n)$,集中的任务用 $T_i = (A_i, E_i, D_i), i = 1, 2, 3, \dots, n$ 来表示。其中, A_i 表示任务的到达时间,即任务被启动且准备执行的时间; E_i 表示任务需要的执行时间; D_i 表示任务的截止时间,即任务在这个时间需要执行并产生结果。

定义2 任务 $T_i = (A_i, E_i, D_i)$ 的剩余期限 $R_i(t)$ 定义为截止时间 D_i 、到达时间 A_i 与当前时刻 t 之差,即

$$R_i(t) = D_i - A_i - t \quad (1)$$

定义3 任务 $T_i = (A_i, E_i, D_i)$ 的剩余执行时间 $N_i(t)$ 定义为执行时间 E_i 与已完成的执行时间 F_i 之差,即

$$N_i(t) = E_i - F_i \quad (2)$$

根据任务的截止时间、剩余时间和所需时间,可以构造本文所提出的调度算法的优先级结构,优先级结构由 3 部分组成,分别为截止时间的优先级 P_{Di} 、剩余时间的优先级 P_{Ri} 和所需单位时间的优先级 P_{Ni} ,如图 1 所示。

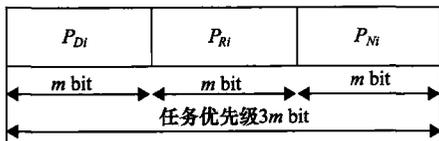


图 1 SREDF 调度算法的任务优先级结构

$$P_{Di}(T_i, t) = \begin{cases} 2^m - 1 - D_i(t), & D_i(t) > N_i(t) > 0, (D_i(t) + t) < D_i(t) \\ 0, & \text{其他} \end{cases} \quad (3)$$

$$P_{Ri}(T_i, t) = \begin{cases} 2^m - 1 - R_i(t), & R_i(t) > N_i(t) > 0, (R_i(t) + t) < D_i(t) \\ 0, & \text{其他} \end{cases} \quad (4)$$

$$P_{Ni}(T_i, t) = \begin{cases} 2^m - 1 - N_i(t), & R_i(t) > N_i(t) > 0, (R_i(t) + t) < D_i(t) \\ 0, & \text{其他} \end{cases} \quad (5)$$

式中, $R_i(t) > N_i(t)$ 表示任务在剩余执行时间内可以

完成; $(R_i(t) + t) < D_i(t)$ 表示任务未到截止时间。优先级计算公式对已经到达且未完成的任务能在剩余的**时间内完成执行才有意义,否则其余的任务视为已无法完成执行的任务。

2 SREDF 调度算法

2.1 算法流程

根据优先级结构来构造如下的调度算法:

① 计算各个任务的 P_{Di} 、 P_{Ri} 和 P_{Ni} , 并选择最高的任务执行;

② 如果有 2 个或 2 个以上任务的 P_{Di} 相同, 则选择 P_{Ni} 最高的任务执行;

③ 如果有 2 个或 2 个以上任务的 P_{Di} 相同, 且 P_{Ni} 相同, 则选择任意一个任务执行;

④ 为了调高系统资源的利用, 在调度任务过程中, 可以提前分析出不可能被执行的**任务, 即在截止期到期前, 任务不够所需的时间片, 导致不可能被执行;

⑤ 取消任务的**条件: (当前时刻 t_i + 需要的时间片 N_i) > 截止期 D_i 或 (剩余时间 R_i < 需要的单位时间 N_i)。

该算法的程序代码如下。

SREDF Schedule Algorithm

//input: new_Task

//output: Highest_Task

Begin

while true if the new task is arrival

then

calculate its P_{Di} , P_{Ri} and P_{Ni} ;

add_Into_Linked_List (new_Task);

end if

sort_By_ P_{Di} _Descending (Linked_List);

If some takes have the same P_{Di}

then

sort_By_ P_{Ni} _Descending (Linked_List);

end if

for j = 1 to Linked_List.length

if (($(t_i + N_i) > D_i$) or ($R_i < N_i$))

// (当前时刻 t_i + 需要的时间片 N_i) > 截止期 D_i 或 (剩余时间 R_i < 需要的单位时间 N_i)

cancel (Task_j);

end for

Highest_Task = Task_{j=1};

Running (Highest_Task);

end while

end

2.2 SREDF 算法分析

截止期错失率^[10] (MDP, missed deadline percentage) 是衡量调度策略优劣的性能指标。MDP 是指系

统中未被调度成功的任务个数与参加调度任务个数之比,它与调度成功率成反比,截止期错失率越高,则调度成功率越低。

考虑有 5 个任务: $T_1(0,4,6), T_2(1,2,9), T_3(1,6,10), T_4(1,5,9), T_5(2,3,7)$, 调度原则是:在相同的相对截止期的条件下,按照最短 CPU 运行期优先调度算法调度;若任务需要的时间片相同,则按照先到先调度的原则。EDF 和 SREDF 算法的执行过程如图 2 所示。SREDF 调度算法从 $t=0$ 到 $t=10$ 时刻,5 个任务的参数 (P_{Ri}, P_{Di}, P_{Ni}) 变化过程如表 1 所示。

表 1 按照 SREDF 调度算法,参数 (P_{Ri}, P_{Di}, P_{Ni}) 变化过程

t	Task 1	Task 2	Task 3	Task 4	Task 5
0	(6,6,4)	-	-	-	-
1	(5,6,3)	(8,9,2)	(9,10,6)	(8,9,5)	-
2	(4,6,2)	(7,9,2)	(8,10,6)	(7,9,5)	(5,7,3)
3	(3,6,1)	(6,9,2)	(7,10,6)	(6,9,5)	(4,7,3)
4	(2,6,0)	(5,9,2)	(6,10,6)	(5,9,5)	(3,7,3)
5	已提交	(4,9,2)	(5,10,6)*	(4,9,5)*	(2,7,2)
6	-	(3,9,2)	已取消	已取消	(1,7,1)
7	-	(2,9,2)	-	-	(0,7,0)
8	-	(1,9,1)	-	-	已提交
9	-	(0,9,0)	-	-	-
10	-	已提交	-	-	-

如图 2 所示,EDF 算法的调度结果为任务 T_1 和 T_5 得到执行,其他的 3 个任务 T_2, T_3 和 T_4 错过了截止期,截止期错失率 $MDP = 3/5 = 0.6$ 。

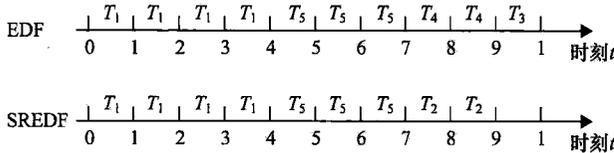


图 2 EDF 和 SREDF 算法的调度结果

SREDF 算法的调度结果为任务 T_1, T_2 和 T_5 得到执行,其他的 2 个任务 T_3 和 T_4 错过了截止期,截止期错失率 $MDP = 2/5 = 0.4$ 。

3 仿真实验分析

本文所用的硬件环境为 LPCXpresso 及实验板 LPC1769。设置了一组任务,记为 t_1, t_2, \dots, t_n , 在仿真过程中取 $n=10$ 和 $n=20$,且 $R(x,y)$ 表示在 x 和 y 之间的均匀随机整数,共仿真 10 批,对每组任务分别采用 EDF 和 SREDF 算法,并统计每批次 MDP 的平均值,执行结果如图 3~图 5 所示。

在图 3 的仿真过程,取 $n=10, A(0,10), D(10,30)$,由图 3 可知,与 EDF 算法相比较,通过 SREDF 算法调度,在这 10 批的测试中,大部分的任务组的截止

期错失率都得到降低。

在图 4 的仿真过程,取 $n=20, A(0,10), D(10,30)$,由图 4 可知,随着任务数的增加,截止期错失率也有所增加,通过 SREDF 算法调度的 MDP 比 EDF 算法调度的 MDP 低。

在图 5 的仿真过程,取 $n=20, A(0,10), D(10,40)$,由图 5 可知,随着截止期的随机数取值范围的增加,提高了任务调度的成功率。

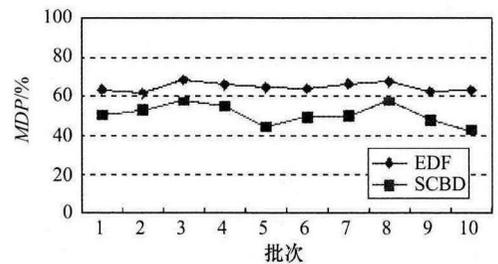


图 3 MDP 的平均值,取 $n=10, A(0,10), D(10,30)$

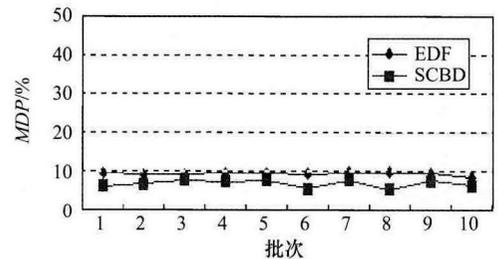


图 4 MDP 的平均值,取 $n=20, A(0,10), D(10,30)$

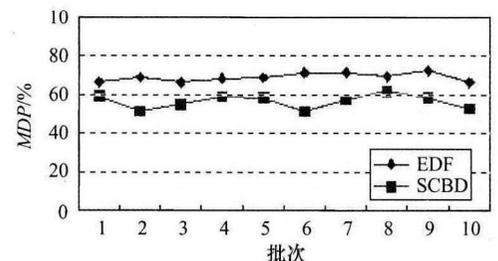


图 5 MDP 的平均值,取 $n=20, A(0,10), D(10,40)$

4 结束语

考虑了任务的截止期和 CPU 运行期这两个特征参数来设计任务的优先级,每个不同的任务有不同的优先级,即一一对应。综合考虑了两个特征参数,使得截止期越早或 CPU 运行期越短时,任务的优先级也越高。仿真结果表明,通过综合两个特征参数来设计任务优先级的调度算法,考虑任务间的重要程度,以及系统完成的任务数量,从而提高了调度任务的成功率。该算法适合于实时性要求较高且完成任务数量多的领域,如应急通信、环境监测,以及交通控制等领域,能保证在规定时间内及时完成任务,提高系统的可靠性。

(下转第 108 页)

getdate())”。



图2 数据查询前面板

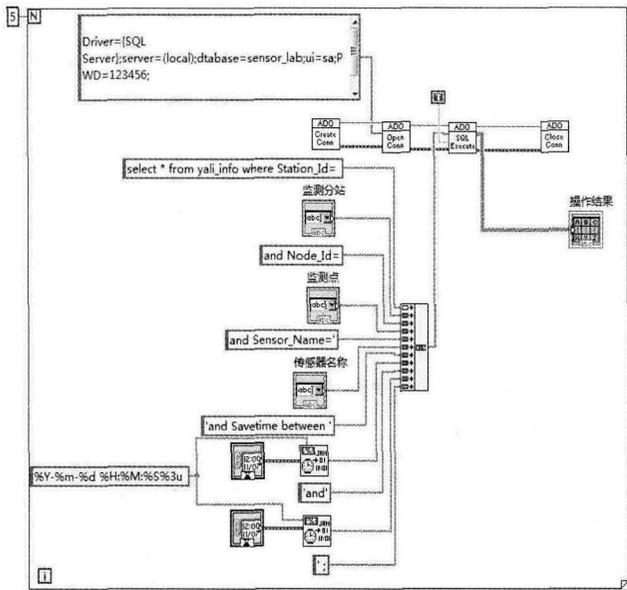


图3 数据查询程序图

删除、更新语句也是按照 SQL 语句实现的,例如删除7号监测点的压力数据,其SQL实现为“delete

from yali_info where Node_Id=7”。将3号监测点归为2号监测分站,即“update yali_info set Station_Id=2 where Node-Id=3”。

5 结束语

超高水充填管道在线自动监测系统基于 LabVIEW 软件平台,借助 LabSQL 工具包实现了与 SQL Server 数据库的连接,并实现了工作人员对数据的管理,满足了用户需求。实践证明,LabSQL 免费工具包具有方便、简捷、易理解的优点,不需要精通 ActiveX 技术和繁琐的编程语句,降低了开发人员的开发难度,节省了开发成本。

参考文献:

- [1] 刘胜,张兰勇,章佳荣,等. LabVIEW 2009 程序设计[M]. 北京:电子工业出版社,2010:289-290.
- [2] 冯光明,王成真,李凤凯,等. 超高水材料开放式充填开采研究[J]. 采矿与安全工程学报,2010,27(4):453-457.
- [3] 马广兴,侯尚武,白长江. 超高水材料充填系统的优化[J]. 中州煤炭,2012(4):59-60.
- [4] 徐小华,贺斌. LabVIEW 中访问数据库几种不同方法的探讨[J]. 中国科技信息,2010(7):122-128.
- [5] 王业,王聪,许加庆. 基于 LabVIEW 与 SQL Server 的心电数据库设计[J]. 计算技术与自动化,2013,32(2):128-132.
- [6] 张蒲生. 数据库应用技术[M]. 北京:机械工业出版社,2008:20-25.
- [7] 张悍东,纪文志. 数据采集系统中的 LabVIEW 数据库访问技术[J]. 工业仪表与自动化装置,2009(4):63-66.
- [8] 唐亚鹏,侯媛彬. 基于 LabVIEW 的实践教学平台与 Access 数据库的开发[J]. 计算机技术与发展,2011,21(5):219-222.
- [9] 李文涛,曹彦红,卜旭芳,等. LabVIEW 数据库访问技术的实现及应用[J]. 工矿自动化,2012(3):69-72.

(上接第80页)

参考文献:

- [1] 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,2004,15(6):799-814.
- [2] Balbastre P, Ripoll I, Crespo A. Minimum deadline calculation for periodic real-time tasks in dynamic priority systems[J]. IEEE Transactions on Computers,2008,57(1).
- [3] 李琦,巴巍. 两种改进的 EDF 软实时动态调度算法[J]. 软件学报,2011,34(5):943-950.
- [4] Leontyev H, Anderson J H. Generalized tardiness bounds for global multiprocessor scheduling[J]. Real-time System, 2010,44(1/2/3):26-71.
- [5] Buttazzo G, Spuri M, Sensini F. Value vs. deadline schedu-

ling in overload conditions[C]//Proceedings of 16th IEEE Real-Time Systems Symposium. 1995:90-99.

- [6] 任小西,赵公怡. 基于动态抢占阈值的 LSF 调度算法[J]. 计算机工程,2012,38(4):275-277.
- [7] 王永炎,王强,王宏安,等. 基于优先级表的实时调度算法及其实现[J]. 软件学报,2004,15(3):360-370.
- [8] 夏家莉,陈辉,杨兵. 一种动态优先级实时任务调度算法[J]. 计算机学报,2012,35(12):2685-2695.
- [9] 李琦,巴巍. 两种改进的 EDF 软实时动态调度算法[J]. 计算机学报,2011,34(5):943-950.
- [10] 金宏,王宏安,王强,等. 一种任务优先级的综合设计方法[J]. 软件学报,2003,14(3):376-382.