

基于仿真方法的任务集可调度性判定工具

徐建华^{1,2}, 李 允³

(1. 西南交通大学信息科学与技术学院, 成都 610031; 2. 北京科银京成技术有限公司成都研发中心, 成都 611731;

3. 电子科技大学计算机科学与工程学院, 成都 610054)

摘 要: 针对含有纯周期任务的任务集可调度性判定问题, 提出一种基于仿真方法的任务集可调度性判定工具。通过设定时钟变量, 模拟任务调度过程中的系统时钟, 在时钟变量值增长的过程中, 根据任务优先级从高到低的顺序, 分析各个任务的截止时间限, 判定任务集的可调度性。实例分析证明了该工具的可调度性判定结果正确。

关键词: 任务集; 可调度性判定; 时钟变量; 周期任务

Task Set Schedulability Test Tools Based on Simulation Method

XU Jian-hua^{1,2}, LI Yun³

(1. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China;

2. Chengdu Research and Development Center, Coretek System Co., Ltd., Chengdu 611731, China;

3. School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu 610054, China)

【Abstract】 Aiming at the schedulability test problem of task set which only contains periodic tasks, this paper proposes a simulation-based schedulability test tool for task set. It simulates the system clock in the process of scheduling by setting clock variable. In the process of this clock variable growth, according to the task priority from high to low sequence, it analyzes each task deadline constraint, judges the single task schedulability. Results of the tool are proved to be correct with schedulability test cases.

【Key words】 task set; schedulability test; clock variable; periodic task

DOI: 10.3969/j.issn.1000-3428.2012.02.079

1 概述

在调度系统中, 如果一个任务在每次到达后都能在其截止时间限内执行完成, 则判定该任务可调度, 否则该任务不可调度。如果任务集中的所有任务都能在各自的截止时间限内完成, 则判定任务集可调度, 否则任务集不可调度。在现有的可调度性判定方法中, 以基于仿真方法和基于形式化方法的判定结果最为精确。现有主流的可调度性判定工具大多基于形式化方法实现, 如 TIMES^[1]是基于时间自动机方法的可调度性判定工具, 能穷举任务集中任务调度的所有情况, 判定结果精确, 但由于其通过时间自动机来构造任务模型, 存在状态空间爆炸问题, 因此判定能力不高, 不能分析含有较多任务数任务集的可调度性。为此, 本文针对含有纯周期任务的任务集可调度性判定问题, 提出一种基于仿真方法实现的可调度性判定工具。

2 任务调度过程

2.1 调度模型限定

本文研究的调度模型有如下限定:

- (1) 在被分析的任务集中, 所有任务都是周期任务, 且任务的周期值等于各自的截止时间。
- (2) 任务具有硬实时系统的强时间限定。
- (3) 各个任务相互独立, 任务的分派到达不依赖于其他任务, 仅依赖于任务的周期。
- (4) 任务是可抢占的, 且在同一处理器下运行。
- (5) 只针对时间限分析, 默认任务所需的其他非时间资源满足要求。

2.2 理论依据

基于上述限定条件, 任务集中所有任务都是周期任务, 且各个任务的分派到达仅依赖于其周期, 因此, 任务第 1 次到达以后的所有到达时刻都是确定的, 用设定系统时钟变量来模拟调度过程的方法是可行的。此时, 如何确定时钟变量的取值范围成为判定算法正确与否的关键。

假设任务集 S 含有 n 个周期任务, 设定 n 个任务都在 0 时刻到达, 这 n 个任务下一次同时到达的时刻为此 n 个任务周期值的最小公倍数的时刻。 n 个周期任务的到达在整体时间轴上呈周期性, 且周期值为所有任务周期值的最小公倍数, 文献[2]给出了相关证明。

由上述结论可知, 含有纯周期任务的任务集在调度过程中, 所有任务的到达整体上呈周期性, 在确定的调度策略下, 任务间的抢占关系是确定的。因此, 任务集中所有任务在调度过程中的到达、抢占、执行、完成是一个确定的路径且整体上呈周期性, 周期值为所有任务周期值的最小公倍数。在用设定系统时钟变量的方法来模拟任务调度过程时, 只需要分析一个周期时间区域内的任务集中各个任务的截止时间限满足情况, 便可判定任务的可调度性, 从而判定任务集的可调度性。

基金项目: 国家自然科学基金资助项目(90718019); 国家“863”计划基金资助项目(2007AA010304)

作者简介: 徐建华(1984—), 男, 硕士研究生, 主研方向: 嵌入式系统开发; 李 允, 副教授

收稿日期: 2011-07-04 **E-mail:** xujh2011@163.com

2.3 仿真过程

定义变量 X 为系统时钟，该工具的判定过程是：设置系统时钟 X 从 0 时刻开始计时，任务集中所有任务都在 0 时刻到达，以步长为 1 个时间单位增加系统时钟值，模拟任务集的调度过程，直到时钟值增长到所有任务最小公倍数时刻。在模拟过程中系统时钟每增加一个时间单位，根据优先级抢占的情况更新任务的执行时钟和响应时钟，更新任务的当前状态，根据任务完成条件和任务超时条件判定任务的调度性，从而判定任务集的调度性。

3 工具整体结构

调度性判定工具的结构框图如图 1 所示。

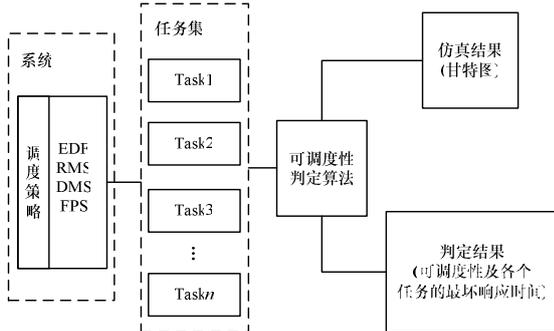


图 1 可调度性判定工具的结构框图

该工具支持的 4 种调度策略为：单调速率调度策略(Rate-monotonic Scheduling, RMS)，短截止时间限优先策略(Deadline-monotonic Scheduling, DMS)，固定优先级调度策略(Fixed-priority Scheduling, FPS)，相对短截止时间限优先策略(Earliest Deadline First, EDF)。其中，前 3 种为静态优先级抢占策略，最后一种为动态优先级抢占策略。该工具能自动提取用户输入的任务参数及调度策略，调用相应的可调度性判定算法，自动、准确、快速地给出任务的调度性判定结果及任务集中所有任务的最坏响应时间。该工具的仿真模块实现了以甘特图的方式单步或连续地仿真显示任务集的调度过程，便于验证工具判定结果的正确性，由于篇幅有限，仿真模块中甘特图的单步、连续显示的具体实现，不再详细描述。

4 工具实现及性能分析

4.1 任务模型

定义 $T = \{t_1, t_2, \dots, t_n\}$ 为一个含有 n 个周期任务的任务集。集合中的任务用 $t_i = (C_i, T_i, D_i, L_i, E_i, W_i)$ 来表示，其中， C_i 表示任务 t_i 的执行时间； T_i 表示任务 t_i 的周期； D_i 表示任务 t_i 的截止时间； E_i 表示任务 t_i 的执行时钟，只有当任务处于运行状态时，执行时钟才开始计时； W_i 表示任务 t_i 的响应时钟，当任务请求到达时，响应时钟开始计时，直到任务执行完成； L_i ：任务 t_i 的当前状态，每个任务在调度过程中的 4 种状态定义如下：

- (1)Suspend: 挂起状态(任务的初始状态)；
- (2)Ready: 就绪状态，此时任务的响应时钟开始计时；
- (3)Running: 执行状态，此时任务的执行时钟和响应时钟都要计时；
- (4)Error: 超时状态，当任务的响应时间超过截止时间限时，任务转向 Error 状态。

任务状态之间的迁移过程如图 2 所示，为叙述方便图中默认编号小的任务优先级高。与文献[3]提出的实时调度框架相比，本文描述的系统模型和任务模型更加详细具体，能更

直观地描述任务模型和具体的调度模型。

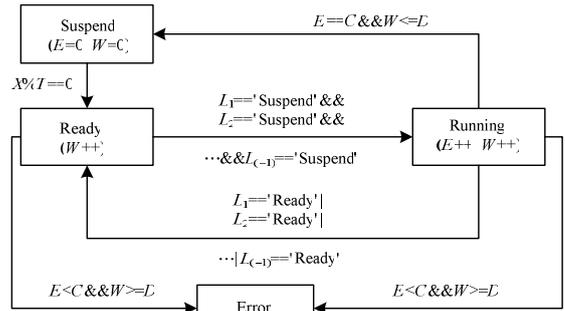


图 2 任务状态转换图

4.2 判定算法

设定系统时钟 X 从 0 开始计时，所有任务在 0 时刻到达，由上述纯周期任务到达执行呈周期性可知，模拟调度过程所用到的系统时钟只需设定为 $[0, num]$ 区域即可，其中， num 为任务集中所有任务周期值的最小公倍数。系统时钟 X 以步长为 1 个时间单位递增，模拟任务集的调度过程。在模拟过程中系统时钟每增加一个时间单位，根据优先级抢占的情况更新任务的执行时钟和响应时钟，更新任务的当前状态，根据任务完成条件和任务超时条件判定任务的调度性，从而判定任务集的调度性。

为优化调度过程，该判定算法优先判定任务集中高优先级任务的调度性，因此，在判定之前须按用户选取的调度策略把任务集中的所有任务按优先级从高到低的顺序排列。在静态优先调度策略下，任务在调度过程中的任何时刻各个任务的优先级固定不变，因此，整个调度过程中只需在初始时刻将任务集中的任务按优先级高低进行排序。动态优先级调度策略，如本文支持的 EDF 调度策略，在任务的调度过程中，任务间的优先级会发生动态变化。在实现过程中，每当系统时钟 X 增加一个时间单位，更新任务的执行时钟和相应时钟后，重新将任务集中的所有任务按更新后的优先级计算结果进行排序，从而保证了每一分析时刻都能按照优先级从高到低的方法判定任务的调度性，从而判定任务集的调度性。文献[4-5]也分别对 EDF 及 RMS 调度算法做了研究。

本文工具的判定算法伪代码如下：

```

bool isScheduler=true;
Sort(TaskSet); //按优先级高低顺序将任务排序
for(int X=0;X<=num;X++)
{ if(!isScheduler)//不可调度则停止判定
break;
for(int i=0;i<taskNum;i++)
{if(Li=="Suspend"){
Ei=0; Wi=0;
if(X%Pi==0){Li="Ready"; //任务分派到达
Wi++;
if(L0=="Suspend"&&...&&L(i-1)=="Suspend")
{Li="Running";
Ei++;}}
else if(Li=="Ready"){
Wi++; //任务处于就绪状态，响应时钟计时
if(L0=="Suspend"&&...&&L(i-1)=="Suspend")
{Li="Running"; //处于执行状态，执行时钟计时
Ei++;}}
else if(Li=="Running"){
Wi++;

```

```

if(L0=="Suspend"&&...&&L(i-1)=="Suspend")
{Li="Running";
 Ei++;}
else if(L0=="Ready"||...||L(i-1)=="Ready")
 {Li="Ready";}
for(int h=0;h<taskNum;h++)
{if(Ei==Ci&& Wi<=Di)
{Li="Suspend";
 Ei=0; //任务完成, 执行时钟和响应时钟清零
 Wi=0;}
else if(Ei<Ci && Wi==Di)
{Li="Error"; //任务超时, 进入 ERROR 状态
 isScheduler=false;
 break;}}
if(strategy=="EDF"){
 Sort(TaskSet);} //动态更新优先级, 重排任务
}

```

上述判定算法的时间复杂度为 $O(taskNum \times num)$, 其中, $taskNum$ 为任务集中的任务数目; num 为任务集中所有任务周期值的最小公倍数。

4.3 性能分析

在静态优先级抢占策略和动态优先级抢占策略下, 针对本文的可调度性判定工具的判定性能问题, 分别以 10 种不同的任务数目, 设置 10 组不同的测试用例。取随机产生的 20 组任务集的判定耗时的平均值作为每组测试用例的测试结果, 并设定所有随机选取的任务集中的所有任务周期值的最小公倍数的范围为 $0 < num < 10^7$ 。测试结果见图 3, 描述了含有不同任务数目的任务集可调度性判定的平均耗时。

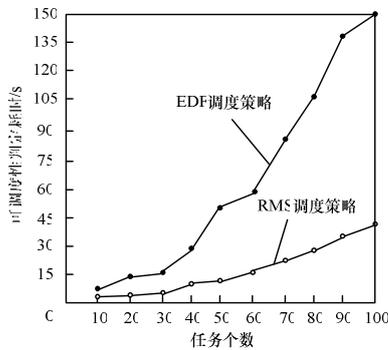


图 3 本文工具的性能测试结果

5 实例分析

设系统时钟从 0 时刻开始, 所有任务都在 0 时刻到达, 任务集中的任务属性参数如表 1 所示。

表 1 任务属性参数

任务名	周期	执行时刻	截止时刻
Task1	3	1	3
Task2	4	1	4
Task3	6	2	6

图 4 描述了上例中任务集在静态优先级抢占策略 RMS 调度策略下的可调度性判定结果和仿真过程。图 5 描述了任务集在动态优先级抢占策略 EDF 调度策略下的可调度性判定结果, 及任务调度的仿真过程。对比相同实例下主流调度分析工具 TIMES 的分析结果可知, 本文的判定工具对该实例中的任务集的可调度性结果、各个任务的最坏响应时间及仿真模块的甘特图显示结果都准确无误。图 4 中的仿真结果显示, 在整个调度过程中 3 个任务的优先级不变, Task1 的优先级最高, Task2 次之, Task3 最低。图 5 中的仿真结果

显示了不同时刻下各个任务的优先级变化情况。图 4、图 5 中存在的共性是, 任务集中的 3 个周期任务的到达执行情况以其周期值最小公倍数 12 为周期重复调度过程。

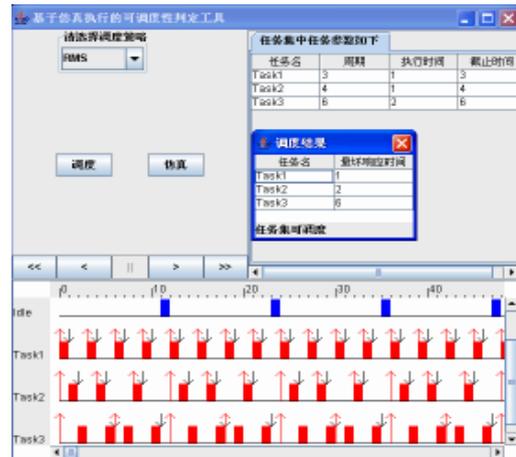


图 4 RMS 调度策略下任务集的判定结果

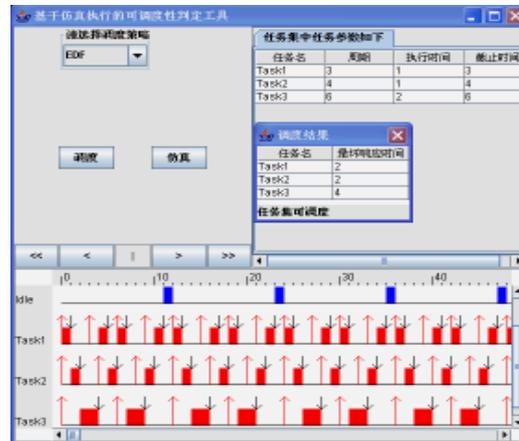


图 5 EDF 调度策略下任务集的判定结果

6 结束语

本文提出一种基于仿真方法实现的可调度性判定工具, 阐述了工具的核心实现过程和性能测试结果。实例证明, 该工具能自动、快速、准确地判定任务集的可调度性, 同时分析出所有任务的最坏响应时间, 并能单步或连续地以甘特图显示的方式模拟任务集的调度过程。下一步工作将研究多实例情况下的任务集可调度判定算法及基于分区操作系统的可调度性判定问题。

参考文献

- [1] Amnell T, Fersman E, Mokrushin L, et al. TIMES: A Tool for Schedulability Analysis and Code Generation of Real-time Systems[C]//Proc. of FORMATS'03. Marseille, France: [s. n.], 2003.
- [2] Leung J Y T, Merrill M L. A Note on Preemptive Scheduling of Periodic, Real-time Tasks[J]. Information Processing Letters, 1980, 11(3): 115-118.
- [3] 金国哲, 陈蜀宇. RTSIM: 一种实时调度模拟框架[J]. 计算机工程, 2010, 36(6): 278-280.
- [4] 萧伟, 冯治宝, 应启夏. 改进型 EDF 调度算法的研究与实现[J]. 计算机工程, 2009, 35(18): 231-233.
- [5] 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.