

嵌入式系统同步进程的竞态条件分析与推理学习方法

张晶 潘有顺

(昆明理工大学云南省计算机技术应用重点实验室 昆明 650500)

(昆明理工大学信息工程与自动化学院 昆明 650500)

摘要 针对嵌入式系统同步进程的竞态条件问题,提出了一个竞态条件分析与推理模型,包括竞态依赖集、竞态协作图和竞态条件数组三要素。其中,模型分析并推理嵌入式系统中具有竞态关系的进程,生成竞态依赖集,定义竞态协作图来描述同步进程竞态条件关系,设计竞态条件数组来存储进程竞态逻辑推理关系以便进一步学习分析。该分析与推理学习方法提高了分析效率,具有一定的实用价值。

关键词 嵌入式系统,竞态条件分析与推理模型,竞态依赖集,竞态协作图,竞态条件数组

中图法分类号 TP316.2 **文献标识码** A

Analysis and Reasoning of Race Condition in Embedded System Synchronization Process

ZHANG Jing PAN You-shun

(Computer Technology Application Key Lab of Yunnan Province, Kunming University of Science and Technology, Kunming 650500, China)

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

Abstract Because race condition in embedded system synchronization process may arise conflict, this paper proposed a race condition analysis and reasoning model including race dependency set, race cooperation graph and race condition array. Race condition model analyzes synchronization processes with race relation in embedded systems, reasons race condition among processes, and generates race dependency set. Race cooperation graph describes synchronization processes and their race condition reasoning relationship. Race condition array is designed to save process race condition reasoning relationship for further study. The method proposed in this paper improves analysis efficiency and has practical value.

Keywords Embedded system, Race condition analysis and reasoning model, Race dependency set, Race cooperation graph, Race condition array

目前有关竞态条件的研究主要以系统中并发任务为对象,关于系统进程同步竞态条件的研究很少。在多道程序环境下,通常程序是不能参与并发执行的,否则它们将失去封闭性且结果不可再现。进程是系统进行资源分配与调度的基本单位,并发进程间能否有效地共享资源与相互协作,完全取决于进程的同步处理。因此,本文将从进程角度分析嵌入式系统中的竞态条件。

进程竞态条件是指两个或多个进程对共享资源进行访问时最终的结果取决于这些进程的执行顺序。嵌入式系统通常都为多任务操作系统,允许多个进程并发执行。当多个进程同时对一个系统资源进行访问时,由于进程间的异步性,可能引起进程死锁,进而导致操作系统崩溃。因此,合理地分析同步进程的竞态条件和逻辑关系成为嵌入式操作系统研究的主要问题。

1 竞态条件研究现状

目前主要有静态分析和动态分析两种方法用于多任务竞

态条件的分析。静态分析法^[1,2]是指通过直接分析嵌入式系统中的程序执行逻辑,在所有执行路径中分析可能存在的竞态条件;动态分析法^[4-6]是指在软件运行期间获取程序执行路径的有效信息,在此基础上进行分析,推导出嵌入式系统多任务的竞态条件。静态分析方法不占用系统执行时间,能穷举推理出系统中所有可能的竞态条件,但有些竞态条件与系统实际运行环境与配置不一致,不具有实用价值,而且会导致竞态条件空间的指数级增长。而动态分析方法虽然状态空间规模受限,推导获得的竞态条件更具有针对性,但一个关键不足之处是占用系统执行资源,影响程序执行效率。

除了理论分析法外,竞态条件分析与推导机制还有图示法^[7-11],主要图形工具有 POEG 图^[10]、消息序列图^[11]与时空图^[7,8]等。这些工具以图形的方式描述系统执行逻辑,分析任务竞态条件,其中时空图如图 1(a)所示,通过分析多任务间的通信时序关系逻辑推导嵌入式系统的竞态条件,但系统任务间的时序约束难以直接表明任务竞态条件逻辑和存储处理机制。

到稿日期:2013-05-20 返修日期:2013-07-29 本文受国家自然科学基金项目(61263017),云南省自然科学基金项目(2012FB137, 2011 FZ060)资助。

张晶(1974—),男,博士,教授,硕士生导师,CCF 会员,主要研究方向为实时与嵌入式软件、物联网软件建模与设计、信息物理融合系统, E-mail: zhangji0548_cn@sina.com;潘有顺(1977—),男,硕士生,主要研究方向为实时与嵌入式软件的建模与仿真。

考虑到时空图在表达嵌入式系统中竞态关系方面的不足,近期,文献[13]针对系统中的竞态条件提出了一种竞态集和竞态条件图。其优点是可以用竞态集存储系统中与竞态条件直接相关的竞态事件。同时,可以通过竞态条件图把竞态集中的竞态条件表示出来,如图1(b)所示。其缺点是竞态集中除了存储执行事件竞态关系外,还存储了事件间时序偏序关系,占用系统空间过大,影响竞态分析效率。其次,对任务事件的处理对象停留在抽象级别上,无法具体量化或指定。

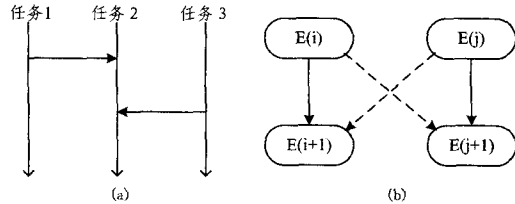


图1 时空图与竞态条件图

2 同步进程的竞态条件分析方法相关定义与定理

嵌入式系统的竞态条件从同步通信原语角度可分为数据竞态条件与一般竞态条件。进一步,一般竞态条件根据操作对象的不同又可具体分为信号量竞态条件与消息竞态条件。数据竞态主要是由不正确的进程同步引起的,而一般竞态条件(消息竞态条件与信号量竞态条件)则因程序难以控制进程访问共享资源的顺序而产生。

定义1(进程) 程序在某个数据集合上的一次执行过程,用 $a(o) = \langle y, t, o \rangle$ 表示。其中: y 表示操作对象类型,包括数据 D 、信号量 S 和消息 M 。 t 表示进程的发生时间。 o 表示进程的操作对象。若 $y = D$, 则 o 为内存中的数据对象;若 $y = S$, 则 o 为内存中的信号量对象;若 $y = M$, 则 o 为内存中的消息对象。

定义2(进程组) 在嵌入式系统中,任意计算机程序一次执行所包含的进程集合称为一个进程组,用 $Q = \langle B, A \rangle$ 表示,其中: $B = \{b_1, b_2, \dots, b_m\}$ 表示嵌入系统中的操作对象集合, m 表示操作对象序号; $A = \{a_1(o), a_2(o), \dots, a_k(o)\}$ 表示本次进程组 Q 中所有进程的集合, k 表示进程序号, o 表示操作对象。

定义3(进程数据竞态) 令 Q_1 和 Q_2 是任一嵌入式系统的2个进程组。若在 Q_1 中进程 $a_1(o)$ 先读取到内存中单元的一个数据,同时在 Q_2 中进程 $a_2(o)$ 对该单元的数据进行改写,则称进程 $a_1(o)$ 和 $a_2(o)$ 之间是数据竞态,如图2所示。

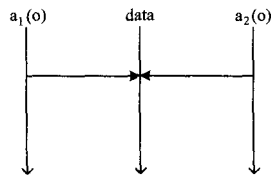


图2 数据竞态时空图

文献[3]针对数据竞态提出了一种有效的 Race Track 方法,它通过对访问内存数据的进程进行动态检测与分析,从而得出系统中的数据竞态条件。

定义4(进程消息竞态^[13]) 令 Q_1 和 Q_2 是任意嵌入式

系统的2个进程组,其存在3个进程 $a_1(o)$, $a_2(o)$ 和 $a_3(o)$, 在 Q_1 中, $a_2(o)$ 先收到 $a_1(o)$ 发送的消息,后收到 $a_3(o)$ 发送的消息;而在 Q_2 中, $a_2(o)$ 先收到了 $a_3(o)$ 发送的消息,后收到 $a_1(o)$ 发送的消息,则称进程 $a_1(o)$ 和 $a_3(o)$ 之间是一个消息竞态,如图3所示。

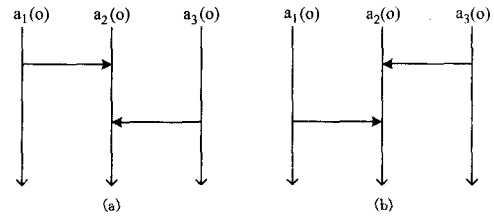


图3 消息竞态时空图

定理1 进程消息竞态一定发生在进程接收消息的时刻。

定义5(进程信号量竞态^[13]) 令 Q_1 和 Q_2 是嵌入式系统中的任2个进程组。若在 Q_1 中进程 $a_1(o)$ 先获取到某个信号量 S , $a_2(o)$ 后获取到同一个信号量 S , 而在 Q_2 中进程 $a_2(o)$, $a_1(o)$ 获取到信号量 S 的次序与 Q_1 相反,则称在系统中进程 $a_1(o)$ 和 $a_2(o)$ 之间具有信号量竞态关系,如图4所示。

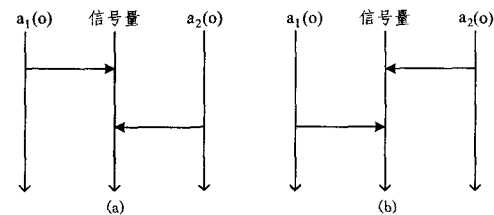


图4 信号量竞态时空图

定理2 进程信号量竞态一定发生在进程获取信号量的时刻。

定义6(进程偏序关系) 在进程组 Q 中,进程之间满足一种时序关系,即进程发生的先后关系,它可用符号“ $--->$ ”来表示。若 $t_1 < t_2$ (即 $a_1(o)$ 先于 $a_2(o)$ 发生),则可表示为 $a_1(o) ---> a_2(o)$ 。反之,若 $t_1 > t_2$,则表示为 $a_1(o) <--- a_2(o)$ 。

定义7(进程竞态关系) 在嵌入式系统中,进程组 Q_1 存在两个进程 $a_1(o)$ 和 $a_2(o)$,其满足定义3、定义4与定义5的条件,则两进程 $a_1(o)$ 和 $a_2(o)$ 在系统中的关系称为竞态关系,用符号“ $->$ 或 $<-$ ”表示。如:进程组 Q_1 的两个进程 $a_1(o)$ 和 $a_2(o)$ 具有竞态关系且 $t_1 < t_2$,则其表示为 $a_1(o) -> a_2(o)$;反之,表示为 $a_1(o) <- a_2(o)$ 。

定理3 进程间的竞态条件关系是一种特殊的偏序关系。

定义8(时序环) 在进程组 Q 中,多个进程之间在时序关系上形成一个闭合环形回路。如: $a_1(o)$, $a_2(o)$, $a_3(o)$, $a_4(o)$ 为进程组 Q 中的4个进程,它们若具有 $a_1(o) ---> a_2(o) ---> a_3(o) ---> a_4(o) ---> a_1(o)$ 的时序关系,就是一个时序环。

定理4 在进程组 Q 中,如果进程间在时序关系上存在时序环,则此竞态条件在系统中是不发生的。

定义9(进程竞态依赖集) 由进程组 $Q = \langle B, A \rangle$ 中与竞

态条件直接相关的进程以及它们之间所具有的竞态关系组成的集合称为竞态依赖集,可表示为二元组 $M\langle C,R\rangle$,其中 $C=\{a_1(o),a_2(o),\dots\}$ 表示具有竞态关系的进程集合, $R=\{->\}$ 表示竞态关系的集合。

定义 10(进程竞态协作图) 用协作图描述进程组 $Q=\langle B,A\rangle$ 的竞态依赖集 $M\langle C,R\rangle$ 中进程的竞态关联关系。其中“○”表示 C 中的进程,“ $->$ ”表示 R 中的竞态关联关系。

为了在计算机中方便地存储进程的竞态条件信息,在此定义了进程竞态条件数组。设竞态依赖集 $M\langle C,R\rangle$ 和进程竞态条件数组 $Array[n][n]$,转换规则如图 5 所示。

$$Array[n][n]=\begin{cases} 1, & (a_i(o),a_j(o))\in R \\ 0, & (a_i(o),a_j(o))\notin R \end{cases}$$

图 5 转换规则

3 同步进程竞态条件分析与推理模型

结合以上相关定义与定理,本文提出了一个针对同步进程的竞态条件分析与推理模型,如图 6 所示。

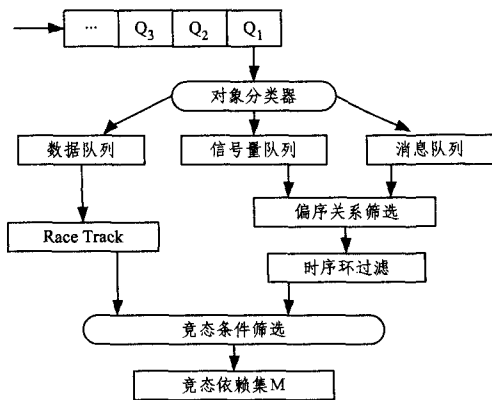


图 6 竞态条件分析与推理模型

该模型以进程组 $Q=\langle B,A\rangle$ 为分析单位。首先,从嵌入式系统中取一进程组 Q ,根据不同的操作对象 B 的类型(即对象分类器),将进程集合 A 归类,划分为消息子队列、数据子队列与信号量子队列。其次,数据子队列利用文献[3]中的 Race Track 方法得出数据竞态关系进程。消息子队列与信号量子队列根据进程时序关系进行偏序关系过滤,依据定义 7 和定理 1、2、4 分别进行竞态条件筛选与时序环判断,分析推断进程组 Q 是否具有消息与信号量竞态条件。最后,将只与竞态条件直接相关的进程及其竞态关系添加到进程竞态依赖集 $M\langle C,R\rangle$ 中。

假设存在一进程组 $Q=\langle B,A\rangle$,经过同步进程竞态条件分析模型,推理得出竞态依赖集 $M\langle C,R\rangle=\{a_1(o)->a_2(o),a_3(o)->a_4(o)\}$,根据定义 10,绘出其对应的进程竞态协作图,如图 7(a)所示。

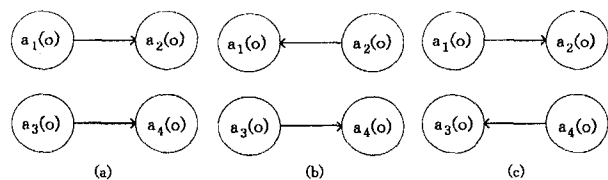


图 7 竞态协作图

与此同时,由定义 7 可知,该嵌入式系统中一定存在其他进程组 Q_1, Q_2 等,它们拥有同样的进程 $a_1(o), a_2(o), a_3(o)$ 和 $a_4(o)$,通过改变进程组 Q 竞态协作图的进程竞态协作方向,可得到嵌入式系统中可能存在进程组 Q_1, Q_2 的竞态协作图,如图 7(b)、(c)所示。因此,根据本文的进程竞态依赖集 M 与其对应的进程竞态协作图,可以推断出系统中其他进程组的同步进程竞态条件。

根据上述的竞态依赖集 M 和竞态条件数组 $Array[n][n]$,转换规则,便可得到对应的竞态条件数组 $Array[4][4]=\{0,1,0,0,1,0,0,0,0,0,1,0,0,1,0\}$ 。

通过上述示例的分析过程,本文所提出的同步进程竞态条件分析与推理模型可以快速推理出在嵌入式系统中运行程序可能出现的直接相关同步进程的竞态条件。

4 实验论证分析

本实验采用 ARM 嵌入式系统开发环境,主要由 3 部分组成:宿主机(安装 Windows XP 的 PC 机)、J-LINK 通用仿真器和目标机(烧入了 Linux 2.6.14 嵌入式操作系统的 zj2440 学习型开发板),用串口线和仿真器将开发板与 PC 机连接起来进行仿真调试。以经典同步问题(生产者与消费者问题)为调试用例,详细说明使用竞态依赖集 M 与竞态协作图来分析嵌入式系统中的同步进程竞态条件。

在生产者与消费者问题中^[12],有生产者与消费者两个进程,用 $a_1(o)$ 和 $a_2(o)$ 表示。有 3 个信号量,分别为 S, S_1, S_2 。 S 实现两个进程对缓冲池的互斥访问, S_1, S_2 表示缓冲池中空缓冲区的数量与满缓冲区的数量。假设在某一个进程组 Q 中,生产者先执行,消费者后执行,则经过竞态条件分析模型得竞态依赖集 $M\langle C,R\rangle=\{a_1(S_1)->a_2(S_1),a_1(S)->a_2(S),a_1(S_2)->a_2(S_2)\}$ 。依据上述所得的竞态依赖集 $M\langle C,R\rangle$,绘制出相关的竞态协作图,如图 8(a)所示。

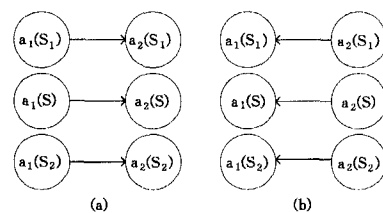


图 8 竞态协作图示例

将上述的竞态依赖集 $M\langle C,R\rangle$ 改变为竞态依赖集 $M_1\langle C,R\rangle=\{a_1(S_1)\leftarrow a_2(S_1),a_1(S)\leftarrow a_2(S),a_1(S_2)\leftarrow a_2(S_2)\}$,则推出这样的进程组 Q_1 ,其执行顺序为消费者先执行,生产者后执行。与其对应的竞态协作图如图 8(b)所示。

结束语 本文针对嵌入式系统进程同步过程中出现的竞态条件问题,深入研究竞态条件情况下进程之间的复杂关联逻辑,设计出一个有效的进程竞态条件分析与推理模型,并利用离散数学中集合及其元素的偏序关系和操作系统中多道程序流水线技术的前趋图方法,总结出可描述进程竞态条件信息的竞态协作图、竞态依赖集与竞态条件数组。实验结果表明,本文的分析与推理学习方法提高了分析效率,减少了系统

的存储空间,具有一定的实用价值。

参考文献

- [1] Engler D, Ashcraft K. RacerX: Effective static detection of race conditions and deadlocks [C] // Proceeding of the 19th ACM Symp on Operating Systems Principles (SOSP). New York: ACM, 2003; 237-252
- [2] Blanc N, Kroening D. Race analysis for SystemC using model checking [C] // Proceeding of IEEE/ACM Int Conf on Computer Aided Design. New York: ACM, 2008; 356-363
- [3] Yu Y, Rodeheffer T, Chen W. Racetrack: Efficient detection of data race conditions via adaptive tracking [C] // Proceeding of the 12th ACM Symp on Operating Systems Principles. New York: ACM, 2005; 221-234
- [4] Klein P N, Lu H I, Netzer R H B. Detecting race conditions in parallel programs that use semaphores [J]. Algorithmic, 2003, 35(4): 321-345
- [5] Pozniansky E, Schuster A. Efficient on-the-fly data race detection in multithreaded C++ programs [C] // Proceeding of PPOPP' 03. New York: ACM, 2003; 179-190
- [6] Tai K C. Race analysis of traces of asynchronous message-passing programs [C] // Proceeding of ICDCS'97. Piscataway, NJ: IEEE, 1997; 261-268
- [7] Lamport L. Time, clocks, and the ordering of events in a distributed system [J]. Communications of the ACM, 1978, 21(7): 558-565
- [8] Park M Y, Hai N C T, Jun Y K, et al. Visualization of message races in MPI parallel programs [C] // Proceeding of the 7th IEEE Int Conf on Computer and Information Technology. Piscataway, NJ: IEEE, 2007; 316-321
- [9] H wang G H, Chang S J, Chu H D. Technology for testing non-deterministic client/server database applications [J]. IEEE Trans on Software Engineering, 2004, 30(1): 59-77
- [10] Schaeli B, Gerlach S, Hersch R D. Decomposing partial order execution graphs to improve message race detection [C] // Proceeding of IEEE Int Parallel and Distributed Processing Symp. Piscataway, NJ: IEEE, 2007; 1-8
- [11] Mitchell B. Resolving race conditions in asynchronous partial order scenarios [J]. IEEE Trans on Software Engineering, 2005, 31(9): 767-784
- [12] 汤小丹, 梁红兵, 哲凤屏, 等. 计算机操作系统(第三版) [M]. 西安: 电子科技大学出版社, 2007; 59-61
- [13] 陈艳, 徐晓峰, 李晓潮, 等. 实时嵌入式系统的竞态条件及其分析方法研究 [J]. 计算机研究与发展, 2010, 47(7): 1201-1210

(上接第 106 页)

简言之,本文提出的 ABC-HC 算法有较快的收敛速度、较好的全局搜索能力,在解决优化问题上优势明显。

参考文献

- [1] Karaboga D. An idea based on honey bee swarm for numerical optimization [R]. Technical Report-TR06. Erciyes University, Kayseri, Turkey, 2005
- [2] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm [J]. Journal of Global Optimization, 2007, 39(3): 459-171
- [3] Banharsakun A, Achalakul T, Sirinaovakul B. The best-so-far selection in artificial bee colony algorithm [J]. Applied Soft Computing, 2011, 11: 2888-2901
- [4] Li G Q, Niu P F, Xiao X J. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization [J]. Applied Soft Computing, 2012, 12: 320-332
- [5] Kang F, Li J J, Ma Z Y. Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions [J]. Information Sciences, 2011, 181(1): 3508-3531
- [6] Gao W F, Liu S Y. A modified artificial bee colony algorithm [J]. Computers and Operations Research, 2012, 39(2): 687-697
- [7] 毕晓君, 王艳娇. 加速收敛的人工蜂群算法 [J]. 系统工程与电子技术, 2011, 33(12): 2755-2761
- [8] 张新明, 孙印杰. 基于混沌优化的自适应中值滤波 [J]. 电子技术应用, 2007, 33(9): 63-65
- [9] 张新明, 徐久成. 基于混沌理论和支持向量机的人脸识别方法 [J]. 高技术通讯, 2009, 19(5): 494-500
- [10] Zhang X M, Yan L. A fast image thresholding method based on chaos optimization and recursive algorithm for two-dimensional Tsallis entropy [J]. Journal of Computers, 2010, 5(7): 1054-1061
- [11] Alatas B. Chaotic bee colony algorithms for global numerical optimization [J]. Expert Systems with Applications, 2010, 37: 5682-5687
- [12] 罗钧, 李研. 具有混沌搜索策略的蜂群优化算法 [J]. 控制与决策, 2010, 25(12): 1913-1916
- [13] 李志勇, 李玲玲, 王翔, 等. 基于 Memetic 框架的混沌人工蜂群算法 [J]. 计算机应用研究, 2012, 29(11): 4045-4049
- [14] 张新明, 雷冠军, 闫林, 等. 一种新型快速的直接随机优化算法 [J]. 吉林大学学报: 理学版, 2012, 50(4): 750-756
- [15] 纪震, 廖慧连, 吴青华. 粒子群算法及应用 [M]. 北京: 科学出版社, 2009
- [16] Yao X, Liu Y, Lin G. Evolutionary programming made faster [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82-102
- [17] Dong H B, He J, Huang H K, et al. Evolutionary programming using a mixed mutation strategy [J]. Information Sciences, 2007, 177(1): 312-327
- [18] Ji M, Yang H, Yang Y, et al. A single component mutation evolutionary programming [J]. Applied Mathematics and Computation, 2010, 215(10): 3759-3768