文章编号:1001 - 9081(2002)2 - 0014 - 03

可升级的高可信嵌入式系统模型

陈 宇,熊光泽
(电子科技大学 计算机科学与工程学院,四川 成都 610054)

摘 要:针对嵌入式系统升级的基本需求及面临的主要问题,提出一种支持高可信嵌入式系统安全的在线升级的模型。详细分析了该模型的理论基础和结构组成,并指出实现该模型需要解决的问题,及其广泛的应用前景。

关键词:嵌入式系统;升级;可信度;分析冗余
中图分类号:TP311.52 文献标识码:A

AN UPGRADEABLE MODEL FOR EMBEDDED DEPENDABLE SYSTEMS

CHEN Yu, XIONG Guang-ze
(Computer Science and Engineering College, University of Electronic Science and Technology, Chengdu Sichuan 610054, China)

Abstract: In this paper, a new model is present, which can realize embedded system on - line dependable upgrade. It describes the theoretical foundation of the upgradeable model, and analyzes the composition of the upgradeable model. In the end of this paper, the problems are pointed, which should be solved to realize the model.

Key words: embedded system; upgradeability; dependability; analytical redundancy

1 引言

嵌入式系统经过几十年的发展,已经从为专用硬件平台开发专门应用发展到以商用嵌入式操作系统为基础,大量商用应用模块与少量定制应用模块相结合,构成应用系统,以降低系统成本,提高系统可移植性。

作为最佳系统,仅考虑系统构造成本和运行可信度是不够的。面对新技术的不断涌现和对系统功能、性能要求的不断提高,用户必须能够针对需求升级系统,延长系统的寿命,增强系统功能,改善系统性能。对于广泛用于工业和国防领域中的嵌入式系统而言,关闭这些系统以实现升级将造成巨大经济损失和安全问题,因此,必须实现在线升级。

嵌入式系统在线升级必须保证升级时和升级后系统的可信度,即升级并不影响系统正常工作,升级后系统的功能、性能和可信度指标最差不能逊于升级前。系统构成之初经过了详细、完备的可信度测试,因此是可信赖的。然而,对于升级时引入的新模块而言,脱离真实运行环境而仅对单个模块所做的测试往往不足以保证其在系统中能正常运行。因此,必须依靠嵌入式系统自身结构实现对升级时出现的错误和升级后新模块出现的错误的包容,以减小升级的难度,保证升级安全。

2 支持升级的模型的基本要求

针对目前嵌入式系统对于升级的需要,我们认为系统的设计应以一种支持升级的高可信嵌入式系统模型为基础,使系统在构成之初就具有良好的升级特性,使用户无需担心引入未经严格测试的新功能模块会对整个系统的功能、性能造成不良影响。我们认为,可升级的高可信嵌入式实时系统体系结构需要满足以下要求:

1) 支持系统升级的模型,自身是可升级的。用户对于升级的要求在不断发展,系统模型必须可以进化以适应这一要求;

2) 升级不能改变系统现有功能模块的代码,以保证系统在新功能模块出错时可以依靠原有功能模块继续运行。脱离真实运行环境而仅对单个功能模块所做的测试往往不足以保证其在系统中能正常运行,因此,必须保证系统中原有功能模块不被破坏,使系统的可信度和性能不低于原有水平;

3) 该结构中,与升级管理相关的功能模块应该独立于系统的其他功能模块,为系统提供标准监测、容错、系统配置和替换功能,使用户专注于系统功能的实现;

4) 该结构作为中间件存在于操作系统和应用之间,可用于不同的操作系统和硬件平台。

3 模型升级的支撑技术

3.1 嵌入式系统的可信度

在嵌入式系统应用的很多方面,用户都需要可信度的保证。可信度指系统值得用户信任的程度。Burns 和 Welling 指出,实时系统的可信度由六个方面构成,如图 1 所示。

提高可信度的基本手段包括:避免错误(在设计阶段发现系统存在的问题,并加以更正,使尽可能少的的错误进入系统

收稿日期:2001 - 08 - 17 基金项目:国防预研项目基金资助(2000J6.7.1.DZ0206)
作者简介:陈宇(1974 -),男,博士研究生,主要研究方向:实时操作系统、软件容错; 熊光泽,教授,博士生导师,主要研究方向:实时计算机系统、实时软件可靠性评测.

的运行阶段),消除错误(在运行阶段,系统提供安全机制,检测并定位已发生的错误,并消除之),容错(在运行阶段,如果已发生错误的功能模块无法采用错误消除技术消除错误并继续运行,则使用一备用模块替换它),规避错误(在运行阶段,采用某些手段预测可能发生的错误,以便在运行中避开它)。以上四个概念指出一个系统在其生命周期的不同阶段为提高其可信度而采取的基本措施。

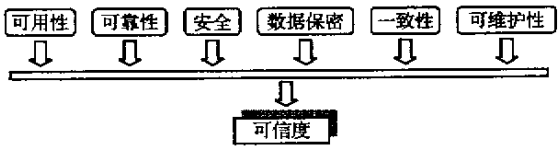

图 1　可信度

3.2　分析冗余

尽管系统依靠异常处理(exception handling),外部监视器(external monitor)和看门狗(watchdog timer)等基本出错检测和处理机制消除错误以提高系统可信度,系统中仍存在以上方法无法解决的错误。在计算机系统中,常采用冗余和投票选择输出的方法,在确认某些功能模块失效后,系统使用备用模块以保证整个系统的正常运行。

冗余主要有三种形式:代码复制、输出一致和分析冗余。代码复制指系统中存在一个功能模块的多个拷贝,其中一个作为主功能模块,当其出错时,系统利用备用模块将其替换。代码复制是冗余中最简单的方法,但同样的模块往往出现同样的问题。在升级时,仅升级少数模块对投票结果没有影响;升级多数模块,若新模块存在设计错误,则可能出现选择错误模块的投票结果。输出一致指代码不同的模块,对于给定的输入,其输出值应该是一致的。由于对于同样的功能采用不同的算法,输出一致的可信度高于代码复制。

然而,在实际工程应用中,冗余模块的输出往往无须一致。对于状态可测和可控的模块,其输出状态可以分为三类:正常、错误和危险。危险指状态处于正常和错误之间,它的出现可能导致错误状态的出现。如果输出结果连续,则系统的告警区域是使系统处于危险状态的最小数值区间。若输出离散,则系统的告警区域是系统从正常到错误的最少危险状态数。对系统各状态间界限的定义是通过权衡系统可信度和性能决定的。由此,我们将各功能冗余模块无需输出一致的系统称为分析冗余系统。例如,由复杂函数的直接运算模块和利用泰勒级数获得的近似多项式运算模块组成的运算系统。虽然对于同样的输入,它们的结果是不同的,但只要能保证泰勒级数的精度,其输出同样能满足系统需要。

3.3　基于 SLP 协议的输出选择

在分析冗余系统中,SLP 协议(Simple Leadership Protocol)用于选择系统的输出:其中,一个功能模块被选为领导者(Leader),其输出用于控制目标系统;其余功能模块被看做表决者(voter),其原理如图 2 所示。领导者和表决者对系统正常状态可能有不同的限定,表决者监视系统状态和领导者的输出,当表决者认为领导者的控制轨迹将导致系统进入表决者认为错误或危险的区域,或领导者未能及时输出结果时,表决者选出的功能模块作为新的领导者控制系统,而过去的领导者将利用本身提供的异常处理句柄或其他手段恢复正常运行,以期在将来某一时刻重新获得控制权。

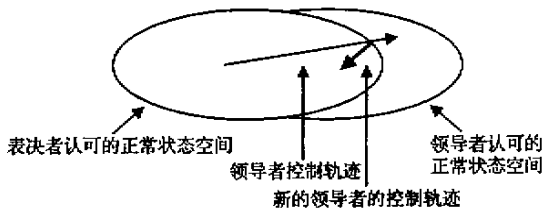

图 2　分析冗余系统结构示意图

SLP 协议具有以下属性:

1) 当 t 时刻领导者的输出导致系统出错时,则在 $t + 2T_{MAX}$ 后,系统将发生控制权转换。T_{MAX} 为系统最长采样时间;

2) 系统能够承受对某一功能模块的主动、恶意攻击;

3) 系统在一个功能模块失效后,能够自动重新配置。例如三重冗余系统在一个模块失效后能自动重新配置成二重冗余系统。

3.4　系统升级的基本原理

在基于 SLP 协议的分析冗余系统中,我们按可信度和性能级别将功能冗余模块按可以分为三等:高可信低性能模块(HALP)、可信度与性能均中等模块(MAMP)、高性能低可信度模块(LAHP)。我们假设模块性能与可信度成反比,且系统允许多个同等级模块并存。因此,系统用一个 LAHP 作为领导者,其余模块作为表决者(包括 HALP、MAMP、LAHP)。当领导者工作正常时,系统获得最佳性能;若其出错时,系统从表决者中选取性能最高者成为新的领导者。可以看出,系统的最差性能和可信度由 HALP 决定。

在系统的运行过程中,新技术的出现使设计新的功能冗余模块成为可能,而脱离真实运行环境仅对单个新模块所做的测试使我们将其定义为 LAHP。本文提出的模型允许系统引入上述新模块并将其指定为领导者,而原有领导者变为表决者中的一员。同样,系统可以使当前领导者退出运行,并从表决者中选择合适的继任者。

3.5　响应时间保证

在嵌入式系统中,引入升级和分析冗余的概念将导致系统开销的增大。因此,系统能否保证响应时间是系统实现的关键。我们认为,以 RM 调度为基础,HALP、MAMP、LAHP 的优先级逐次降低,采用优先级继承协议、资源预留等方法以保证资源的优先访问,可以保证系统中至少有 HALP 能够完成运算,使系统能够及时获得所需控制指令。

4　可升级系统模型分析

在工程实践中,一个可升级的高可信嵌入式系统可能存在多种功能,其存在方式可能是并行,也可能是串行。因此可将每个独立的功能看做分析冗余子系统,其中包含一至多个同样功能的功能冗余模块、一个决策模块、一个替换管理模块和一个数据交换模块。系统由决策模块根据各功能模块的输出和系统状态选择选择合适的领导者,由替换管理模块动态的装载和卸载子系统内的模块。为减少子系统间的耦合,提高子系统的可移植性和软件的重用性和安全性,子系统与外界的数据交换通过数据交换模块实现。

图 3 所示为一个三重功能冗余的分析冗余子系统,由三个功能模块,决策模块、决策规则、替换管理模块和数据交换

模块组成。箭头走向是数据和控制的传输方向。从图中可见,功能模块同外部世界通信只能通过数据交换模块实现。决策模块通过决策规则和各功能模块的输出和系统状态决定系统领导者。替换管理模块实现对功能模块的替换操作。值得注意的是,该分析冗余子系统中的决策模块和决策规则也可以在替换管理模块的控制下实现升级。

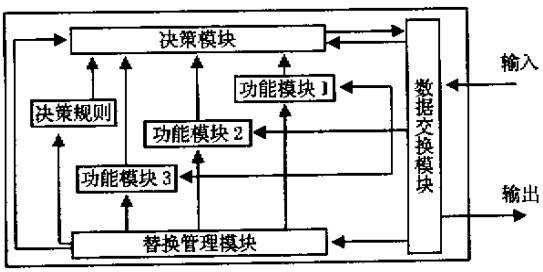

图 3 分析冗余子系统结构简图

4.1 可替换模块和替换管理

冗余子系统中,可替换性是组成系统的每个可替换模块必备属性,是实现系统升级的基本保证。系统组成模块可分为可替换和不可替换两类。可替换模块主要包括系统的功能模块。为升级该子系统的管理功能,相关的管理功能模块也需要具有可替换性。

如图 4 所示,可替换模块由应用程序、替换界面、通信界面和异常处理句柄组成。应用程序根据所需功能的不同,内容差别很大。为使应用程序能从错误中恢复,程序员可以提供必要的异常处理句柄。替换界面实现替换管理模块与本功能模块的信息交流。为减小模块间的耦合,进一步方便升级,模块通过其通信界面实现模块间的交互。替换界面与系统的替换管理模块间的通信也是通过通信界面实现的。

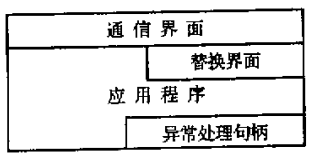

图 4 替换模块结构简图

系统的替换管理模块实现系统功能模块地在线替换。在替换过程中,替换管理模块将引导新模块获得其运行所需信息,并等待新模块进入稳定运行状态后,完成模块的切换。实际的替换包括以下过程:

1) 根据用户提供的信息创建新的可替换模块;

2) 引导新模块获得运行所需信息 新模块开始运行后,其输出由替换管理模块监视,但并不用作系统输出;

3) 当替换管理模块认为新模块的输出以达到系统要求,正式启用新模块,若用户需要,替换管理模块将在适当时刻关闭并删除被替换模块。

当子系统中有多个模块需要同时替换时,可以采用同步替换策略。在所有新模块创建成功后,替换管理模块监视其输出,知道所有新模块都达到正常运行状态,再进行替换操作。如果替换中出现错误,则放弃所有替换操作。

4.2 决策模块和决策规则

决策模块在系统需要时实现输出权的转换,即选择新的领导者。有效的输出总来自领导者。为提高软件的重用性,决策模块的功能与具体应用无关。

决策规则是一系列由用户定义的输出选择规则,可以看做有限状态机,规定了在各种系统状态下,控制权在模块间迁移的规则。

决策模块在决策规则和表决者发出的决策辅助信息的支持下实现输出的选择。为保证系统的可信度,决策模块和决策规则必须是可信的。

4.3 数据交换模块

数据交换模块主要功能包括:

1) 实现分析冗余子系统中,各功能模块间的通信和数据存储;

2) 实现分析冗余子系统与外部世界间的信息交流。数据交换模块是系统内部与外界通信的唯一通道,具有数据过滤器和路由器的功能。

通过这一模块避免了系统内部模块间,系统与外界环境间的直接互操作,减小系统的耦合,提高系统的可移植性,软件的可重用性和安全性。

5 结束语

对于广泛用于工业、国防领域中,处于关键地位,生命周期较长,不能经常关闭以实现系统性能升级的嵌入式系统而言,本文给出的模型使其能够采用新模块在线升级系统而不影响系统可信度。这为提高系统性能,保证系统可信度,延长系统生命周期,降低系统开发成本都具有重要意义。

在目前提出的体系结构基础之上,需要解决的主要问题包括:

1) 对于系统实时特性的形式化描述,以便可以在设计阶段确定系统的可调度性;

2) 模块动态链接的实现。目前虽然某些操作系统支持这一功能,如 Modula III,但作为超越操作系统之上的体系结构,应该找到一种与操作系统无关的方法;

3) 由于数据交换模块是系统中与外界交互的唯一通道,因此高效的通信手段是提高系统效率和实时性的必经之路。

参考文献

[1] Burns A., Welling A. Real – Time System and Programming Languages [M]. Second Edition. Addison – Wesley, 1996.

[2] Joel Sherrill , Jeff Mayes . SAFER : A Scaleable Architecture for Embedded Reliable Real – Time Systems[R]. On – Line Application Research Corporation, Technical Report, OAR – TR – 99 – 183 – 03, June 1999.

[3] Sha Lui, Rajkumar Ragunathan, Gagliardi M. A Software Architecture for Dependable and Evolvable Industrial Computing Systems [R]. Software Engineering Institute, Technical Report, CMU/SEI – 95 – TR – 005, July 1995.

[4] Gagliardi Michael , Rajkumar Ragunathan , Sha Lui . Designing for Evolvability: Building Blocks for Evolvable Real – Time Systems[J]. Proceedings of the IEEE Real – Time Technology and Applications Symposium, June 1996.

[5] Lui C. L., Layland J. W. Scheduling Algorithms for Multiprogramming in a Hard Real – Time Environment[J]. Journal of the Association of Computing Machinery, 1973,20(1).46 – 61.

[6] Walter L . Heimerdinger , Charles B . Weinstock . A Conceptual Framework for System Fault Tolerance [R]. Software Engineering Institute, Technical Report, CMU/SEI – 92 – TR – 33, July 1992.