

构件化可配置实时嵌入式操作系统设计与实现^{*}

施笑安^{1,2} 周兴社¹ 姚华新¹

(西北工业大学计算机学院 西安710072)¹ (空军工程大学电讯工程学院 西安710077)²

摘要 新一代嵌入应用需要自配置和定制操作系统,剪裁掉不必要的系统功能来获得更高的性能和更低的系统开销。软件构件是以可重用为目的设计、封装软件的技术,是操作系统配置管理的基础。提出设计构件化可配置实时嵌入式操作系统的方法:系统功能划分、构件界面划分、配置处理,并依据该理论进行了系统设计与实现,原型系统证明是可配置的,易于用户定制的,也提供了实时嵌入式操作系统所必要的特性。

关键词 构件,可配置,操作系统,嵌入,实时

Design and Implement of Component-Based Configurable Embedded Real-Time Operating Systems

SHI Xiao-An^{1,2} ZHOU Xing-She¹ YAO Hua-Xin¹

(College of Computer Science, Northwestern Polytechnical University, Xi'an 710072)¹

(The Telecommunication Engineering Institute, Air Force Engineering University, Xi'an 710077)²

Abstract New embedded applications need self-configure and customized operating systems, to meet the requirement of specific application and different applying fields. By cutting off the unnecessary parts of the system, it reaches higher performance and requires low consumption of the system. Software component technology is a technique for designing and encapsulating aimed at reuse. It is the foundation of the configuration and management of an operating system. Suppose the design method of the componentized configurable real-time embedded operating system based on the study of contemporary real-time embedded operating system and component technology: defining system function, defining interface, configuration process. It also includes the devise of the configurable real-time embedded operating system. The prototype system is proven to be configured and easy to be customized. Last but not least, it provides the obligatory characteristics of the real-time embedded operating system.

Keywords Component, Configurable, Operating systems, Embedded, Real-time

1 介绍

新一代嵌入应用除了具有传统嵌入式系统具有的专用性强且资源有限、软/硬件紧密融合、多任务实时并发等特点之外,还具有嵌入硬件集成化(如SOC)、嵌入接口多样化、嵌入系统网络化、嵌入应用智能化的新特点,使得传统的实时嵌入计算面临新的挑战。

这些嵌入式应用的每一应用方面都具有其独特性,因此传统的通用操作系统平台对于嵌入式应用并不适用,它很难满足嵌入式应用对于处理器性能、系统开销、实时性及灵活性的各方面要求。随着微内核技术的出现,及操作系统模块化和灵活性的发展,应用可配置操作系统愈来愈广泛应用于实时嵌入式应用中。所谓应用可配置的操作系统的系统主要是指操作系统可根据应用来配置和定制以满足特定应用和特定应用领域的需求,它通过剪裁掉对应用不必要的系统功能来获得更高的性能和更低的系统开销。因此,使用软件构件(简称构件)技术构建可配置实时嵌入式操作系统,是当前实际应用发展的需要。构件并不是一个新的概念,构件技术在软件工程领域已经有多年的应用,以CORBA、COM和JavaBeans等技术为代表,使用构件技术开发的产品已在互联网和分布式系统中有广泛应用。软件构件技术应用于嵌入式系统虽然历史不长,却显示了旺盛的生命力。

本文在对当前实时嵌入式操作系统和构件技术研究的基础上,提出了设计构件化可配置实时嵌入式操作系统的方法,并依据该理论进行了系统设计实现,原型系统证明是可配置的,易于用户定制的,同时也提供了实时嵌入式操作系统所必要的特性。

2 构件化系统设计

软件构件技术是以可重用为目的设计、封装软件的技术。它具有预先定义的接口和明确的上下文依赖关系的软件单元;软件构件是一个预先定义的功能模块,可以重新使用或者重新安排而构成新的软件实体。基于构件的系统可以分为三层,最高层是像Corba、COM和JavaBeans这样大规模的构件。这样的系统通常包括了整个系统和子系统,其接口之间的通讯开销也很高。第二层就是基于构件的具有可配置功能的操作系统,如OSKit。但是OSKit构件只是划分了典型的操作系统功能,它并不支持嵌入式系统。最后一层,细粒度的构件包括系统功能构件、用户提供构件以及应用定制构件。这些构件试图支持低负载的接口,快速中断响应等。但是,目前构件的大小尚无通用的定义,而且不同的系统对于构件划分的粒度也各不相同。这一层主要是指可配置的嵌入式系统。

2.1 构件化系统的功能划分

构件就是可复用的软件组成成分,具备协议指定的接口

^{*} 基金项目:国家自然科学基金(60273086),航天基金(2003CH100001)和西北工业大学博士创新基金资助。施笑安 讲师,博士生,主要研究方向:网络与分布式软件、嵌入式操作系统。周兴社 教授,博士生导师,研究方向为网络与分布式软件。姚华新 硕士生。

以及明确的上下文依赖关系,可被用来构造其它软件。它可以是被封装的对象类、类树、功能模块、软件框架、软件构架(或体系结构)、文档、分析件、设计模式等。从广义上讲,软构件技术是基于面向对象的,以嵌入后马上可以使用的即插即用型软构件概念为中心,通过构件的组合来建立应用的技术体系。狭义上讲,它是通过构件组合支持应用的开发环境和系统的总称。

操作系统级的系统复用开发首先需要我们将操作系统体系结构进行某种合理划分和抽象,使其易于采用构件技术进行架构。如果从功能进行划分,那么一个系统就可以被看作是多个功能模块的组合,但是仅从这一点出发仍然没跳出传统的设计思路,也就不能满足嵌入式操作系统的性能及应用需求,所以我们需要使用新的划分方法来分析嵌入式操作系统。一个系统中所有的功能都具有不同的上下文环境,而总有一些功能具有相似的上下文环境,我们可以根据上下文环境将操作系统功能集合作如下的划分,如图1所示。

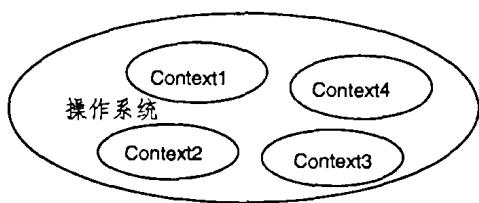


图1 操作系统上下文环境功能划分

其中 Context n 表示具有上下文环境 n 的系统功能的集合,如 Context1中划分的是与嵌入式硬件系统相关并提供与其交互接口的功能模块等。不同集合中的功能模块之间上下文环境差异较大,因此功能模块的接口及其组织形式也就有较大的不同,而同一集合中的功能模块则在接口及组织形式上有较大的相似性,这就使得我们有可能从同一集合功能模块中抽象出一个统一的模块模型,通过设定模块模型通信参数及常数就可以生成特定的功能模块。集合化划分是理清嵌入式操作系统结构的一种方法,基于这一划分我们将其对应到嵌入式操作系统的层次化体系结构,即将同一集合中的功能模块划分到同一层次,而系统本身则具有一个多层次的框架。由于实时嵌入式操作系统本身是面向特定应用领域的、专制的实时系统,具有基本的多任务、文件/设备驱动、支持特定设备及微内核、网络可接入、可伸缩、可配置等功能,所以根据该系统特点较为理想的是4层次体系结构:硬件接口层、核心层、系统层、应用服务接口层。如图2所示。

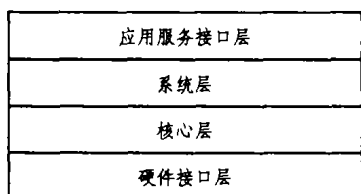


图2 操作系统层次体系结构

·硬件接口层:提供与嵌入式硬件系统交互的接口,该层中的功能模块主要用于与底层硬件的通信与交互,以设备驱动程序为主。

·核心层:即系统的微内核,包括运行一个系统所需的最基本的功能模块,是系统可运行所需功能集合的最小子集,可以运行在多种嵌入式微处理器之上。针对应用提供集中最基

本的操作,如:系统时钟、电源管理、程序装载与运行、进程/线程调度、内存管理等,同时提供一个实时系统所需要的基本要素,如:实现多任务、优先级驱动的紧急优先调度方法、快速现场切换等机制。

·系统层:系统的功能模块化和对象化,提供面向对象的系统资源管理功能,如:内存管理、文件/目录管理、设备管理、网络协议管理等。这一层的架构我们可以根据实际应用的需要选择特定的功能模块或构件,每个系统级模块又提供标准的系统服务的 API 接口,向用户开放,有利于系统功能的扩展。

·应用服务接口层:提供基于系统功能的、面向应用的系统功能调用及服务接口。这一层为用户在特定领域的应用提供了特定服务。

与软件工程中的软件复用技术不同的是,嵌入式系统的复用技术并不需要大量的知识及经验,而最重要的是制定出规范,规范并不限制做什么而是限制如何做以确保构件重用。上面的层次结构很明确地将操作系统进行了分层,基于这个体系结构我们可以制定构件设计规范来指导可重配置软构件的实现,这也有利于第三方对系统性能及应用的扩展。

2.2 构件的界面划分

为了开发基于构件的可配置实时嵌入式操作系统,必须清楚地划分各构件的界限,如果没有适当的方法及正确的方针,那么基于构件开发系统常常很难成功。结合层次划分,需要将系统再次分解为可独立开发、测试并通过通信就可以集成的子系统。通常我们基于功能来划分构件,构件的独立性及其粒度大小是评价构件性能的重要指标。独立性就是一构件与其他构件的依赖性程度,完全独立的构件不依赖于任何其他构件,具有独立的、完善的自包含功能特性,基于这种构件的架构系统非常简单、方便。但是由于这种构件自包含了所有可能用到的功能,所以这样的构件粒度较大而且基于其架构系统的灵活性较差。而粒度小的构件由于其尺寸较小,功能相对简单,进行系统架构就灵活一些,但是正是由于其功能简单,所以构件架构某一功能模块所需的构件就比较多,使得构建系统的工作变得复杂。因此,我们在划分构件的时候就必须把握独立性与粒度大小的合理取舍,以便于高效灵活地开发构件、架构系统。在分层体系结构中,位于不同层次的构件其接口形式及独立性、粒度大小都有所不同,如硬件接口层中的构件主要是与硬件相关的硬件依赖构件,这些构件通常用来驱动硬件并和硬件进行通信,这些构件都和特定硬件相关,因此独立性好但粒度较大;又如系统层中可能包含内存管理构件、网络协议管理构件等,在协议栈中我们又提供了多个独立的子协议——即由多个独立性较好微构件构成。总之,构件的划分要基于功能对象并结合目标系统的应用场合的要求进行划分。

2.3 构件化的配置处理设计

配置处理很少受重视,但它却对系统的性能十分关键。该部分要研究的关键性问题包括:

(1) 构件的定义 使用构件定义语言,构件描述语言实现操作系统的配置。通过使用构件定义语言,配置工具能够支持组件的一致性与相关性检查,从而保证了所选配置在目标系统里正确工作。

构件定义语言与配置工具紧密结合,是操作系统可配置性的关键组成部分。操作系统代码由各个不同的包组成,每个包里至少需要包含一个构件定义语言脚本,每个包的脚本都

包含这个包的配置选项细节以及编译信息。

(2) 构件的选择 引导开发者根据存储空间、实时性、成本、能耗的要求选择合适的构件。可配置性日益成为衡量嵌入式操作系统好坏的重要指标。但是如何为用户提供适当的配置界面,引导用户进行符合自己需求的操作系统配置,却往往不受重视。有些嵌入式操作系统的配置采用直接修改 Makefile 文件的方式,需要用户具备相当的专业知识、对操作系统内核有较深入的了解;有些系统提供了命令行方式的配置界面,界面呆板、命令繁杂,让人望而生畏。我们希望能够为用户提供一个人机交互界面,根据用户应用的需求,提供一组可供选择的解决方案以及直观、生动的显示界面,从而引导用户使用适当的构件构建合适的嵌入式操作系统。

(3) 构件集合的性能分析和测试 分析开发者所选择的构件集合能够满足正确性、实时性(截止期)和其它所需特性。同时嵌入式应用,尤其是应用于复杂环境下的实时嵌入式应用,对于实时性和可靠性的要求尤其突出。嵌入式系统在投入使用前,必需经过严格的性能测试,如果应用实际系统进行测试,一旦失败,必然带来巨大损失。而使用基准测试平台,能够在很大程度上降低测试的风险。利用软件的方法,模拟一个接近真实的环境,通过调节不同参数,可以分析用户所选构件是否能够满足具体应用所需的各项指标。

通过以上关键技术的研究,我们开发了基于构件的可配置嵌入式操作系统,实现为基于构件的可配置的操作系统原型、构件定义语言和配置界面。

3 系统实现

本可配置实时嵌入式操作系统设计,基于开放源码的 e-Cos、OSKit 和 RTEMS 操作系统,运用构件设计的思想,实现了一个构件化可配置实时嵌入式操作系统原型——EMOS, EMOS 以构件库形式提供。此外 EMOS 基于 GNU 工具,提供了完整的嵌入式系统软件开发环境,包括操作系统配置工具、编译环境、模拟环境、下载工具和调试环境,以方便用户进一步进行研究和开发。

3.1 操作系统的软件构件

(1) 轻量级操作系统构件接口设计 我们确立了操作系统构件应实现的功能界面、使用该操作系统构件所依赖的环境和该操作系统构件的测试程序。操作系统的界面具备相当的兼容性和相应的灵活性。该嵌入式操作系统重用构件的主要界面应包含硬件抽象层 HAL, 设备驱动编程界面, 文件系统包界面、应用程序编程接口 API 等。

应用程序接口支持已经存在的嵌入式实时应用编程接口的多种标准,如 IEEE 的实时 UNIX 标准 POSIX, 日本的 uLtron 标准等。实时内核界面应包含多线程支持(线程管理, 线程同步原语), 内存分配原语函数, 设备管理函数, 中断和异常、定时器管理的函数, 标准 C 界面。硬件抽象层(HAL)应用于隔离所有与平台相关的汇编代码, HAL 提供如上下文切换和 I/O 寄存器访问等。这样使实时操作系统核心系统和设备驱动程序可用 C 语言编写, 简化了操作系统的移植。设备驱动编程界面虽无明确主流, 由于目前 LINUX 系统设备驱动比较完整, 嵌入式操作系统将以 LINUX 设备驱动接口为主要构架。文件系统应支持包重用界面, 由 C 库提供的, C 库建立在文件系统支持包基础上。而文件系统支持包含文件名空间管理, 各种文件系统的实现。它们都建立在实时内核界面上。

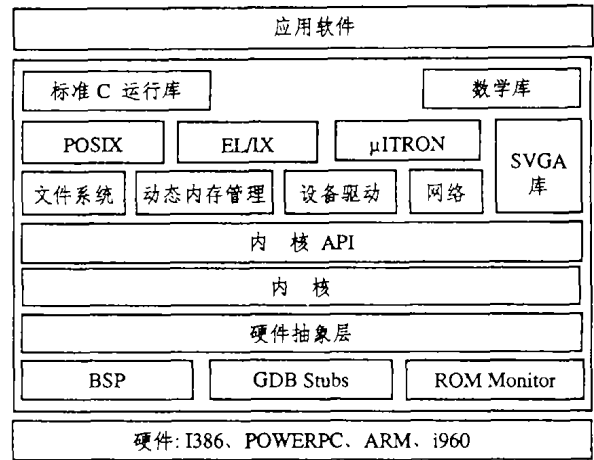


图3 嵌入式操作系统总体结构图

通过轻量级操作系统构件接口设计、构件分类技术,确立该操作系统构件实现功能界面、使用该操作系统构件所依赖的环境和该操作系统构件的测试程序。操作系统的界面具备相当的兼容性和相应的灵活性。该嵌入式操作系统构件的主要界面应包含硬件抽象层 HAL, 设备驱动编程界面, 文件系统包界面、应用程序编程接口 API 等。

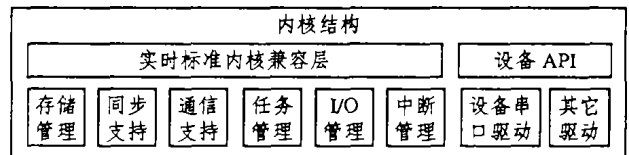


图4 嵌入式操作系统核心结构

(2) 构件分类 对构件之间的相互关系,以及各个构件对系统性能的影响进行分析,按内存大小,执行时间,容错、安全和服务需求(QoS)进行综合分类。并对每一类构件给出明确定义、实现规范,以及清晰的接口定义度量标准,从而达到方便快捷的开发基于构件的嵌入实时系统的目的。

(3) 构件开发和配置信息的保存 以一种开放源码的构件定义语言为蓝本,研究如何描述构件的配置信息,并以现在广泛应用的 XML 语言为基础,开发符合以上研究的构件描述语言,包括构件配置细节,如何将构件引入系统中,以及编译构件信息等。

3.2 构件配置界面的实现

通过对现有语言的改造,我们实现了一种基于脚本的构件描述界面——构件定义语言,类似自然语言的语法,简单易用。通过构件定义语言,提供各构件的配置选项、依赖关系、约束、在线帮助文档的位置等。实现统一的构件描述界面。其结构如图5所示。

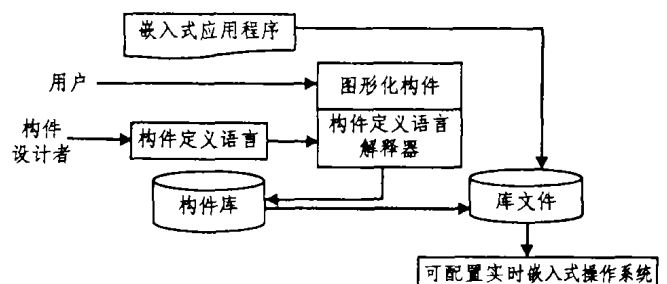


图5 构件配置界面体系

具体实现的嵌入实时操作系统配置环境如图6,包含可

视觉的编译选项管理,定义参数化模块的参数等工作。该可视化配置工具能够控制操作系统配置中的复杂性,极大地增加了系统的应用范围,减轻了开发的复杂度。

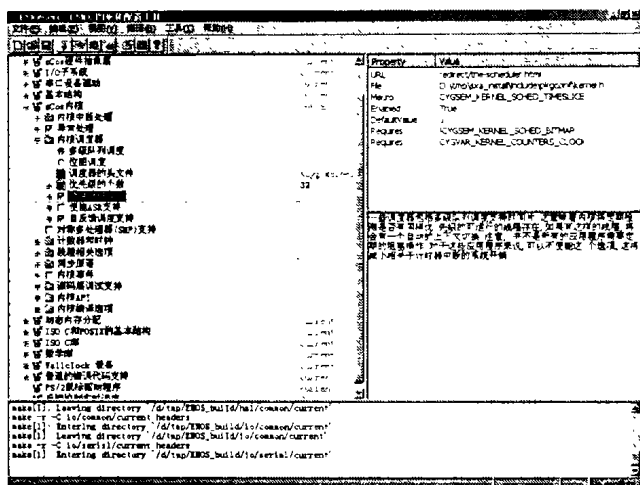


图6 可配置的嵌入实时操作系统

3.3 配置工具的基准测试

我们在配置工具内嵌入一个基准测试程序,验证所选构件的正确性以及检验所选构件是否能满足实际应用的需要。在构件设计时,由设计者提供相应的基准测试,与构件同时封装。在构件配置完成后,通过设置不同的基准参数,模拟嵌入

(上接第196页)

比较(12)和(16),对于同一条曲线利用新方法和 Möbius 变换取得最优参数化时,二者采用的变换是一致的,因而取得一致的最优参数化。

4 实例

一条二次多项式 Bézier 曲线(图1),我们的新方法取得了改进的参数速度曲线(图2)和改进的参数流(图3)。

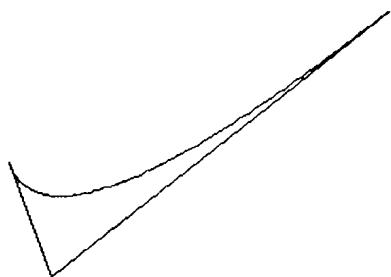


图1 一条二次多项式 Bézier 曲线段

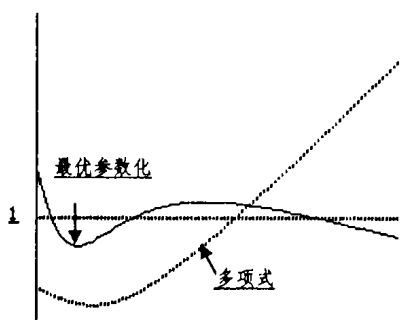


图2 参数速度比较

式应用的实际应用环境,测试构件状态,从而分析配置的合理性。

结论 我们研制的基于构件的可配置嵌入式操作系统具有自主知识产权,同时研制的构件定义语言和操作系统配置测试工具具有一定的创新性。可广泛用于实时嵌入式(如信息家电、掌上电脑、远程传感器等)计算机系统中,根据应用和环境的需要构造嵌入式系统,充分发挥嵌入式系统的功能,并最终降低嵌入式系统的开发和使用代价。

参考文献

- 1 Ciancarini P, Cimato S. Specifying Component-Based Software Architectures. In: Proc. of the ESEC/FSE Workshop on Foundations of Component-Based Systems (FoCBS), Zürich, Sep. 1999. 60~70
- 2 Cygnus eCos - Embedded Cygnus Operating System. Technical White Paper. <http://www.cygnus.com/ecos>, 2003
- 3 Szyperski C. Component Software Beyond Object-Oriented Programming. Addison-Wesley, ACM Press, New York, 2001
- 4 Takada H. ITRON: A Standard Real-Time Kernel Specification for Small-Scale Embedded Systems. Real-Time Magazine, 2000
- 5 Microsoft Corporation. Building the Componentized Operating System in Microsoft Windows CE 2.1. <http://www.microsoft.com/>
- 6 Stewart D B, Volpe R A, Khosla P K. Design of Dynamically Reconfigurable Real-Time Software Using Port-Based Objects. IEEE Transactions on Software Engineering, 1997, 23(12)
- 7 Ford B, Back G, Benson G, Lepreau J, Lin A, Shivers O. The Flux OSKit: A Substrate for Kernel and Language Research. In: Proc. of the 16th ACM Symposium on Operating Systems Principles, Saint-Malo, France, 1997
- 8 Gabber E, Small C, Bruno J, Brustoloni J, Silberschatz A. The Pebble Component-Based Operating System. In: Proc. of the USENIX Annual Technical Conf. Monterey, California, USA, 1999

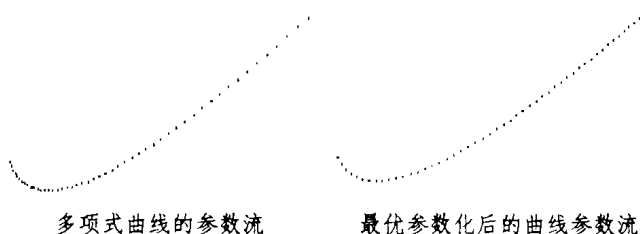


图3 参数流比较

J 从多项式曲线的 1.22 降为最优参数化后的 1.007 ($J=1$ 是弧长参数化)这个结果与 Möbius 变换取得的最优参数化一致。在实际应用的多数情况下,这个参数化与弧长参数化已经很难区分。

结论 利用我们的新方法可以求出多项式 Bézier 曲线的最优参数化。对于同一条曲线新方法求出的最优参数化和 Möbius 变换取得的最优参数化是一致的,且计算比 Möbius 变换简单,通用性强。

参考文献

- 1 Farouki R T, Sakkalis T. Real rational curves are not "unit Speed". Computer Aided Geometric Design, 1991, 8: 151~157
- 2 施法中. 计算机辅助几何设计与非均匀有理 B 样条. 高等教育出版社, 2001
- 3 Farouki R T, Feldman B. Performance analysis of CNC interpolators for time dependent federates along PH curves. CAGD, 2001, 18: 245~265
- 4 Ong B H. An extraction of almost arc-length parameterization for parametric curves. Ann. Number. Math, 1996, 3: 305~316
- 5 Wang F C, Wright P K. Open architecture controllers for machine tools, Part 2: A real time quintic spline interpolator. ASME J. Manufacturing Science and Engineering, 1998, 120: 415~432
- 6 Wang F C, et al. Approximately arc-length parameterized quintic interpolating splines. ASME J. Mech. Design, 1999, 121: 430~439
- 7 Farouki R T. Optimal parameterizations. Computer Aided Geometric Design, 1997, 14: 153~168
- 8 梁锡坤. Bernstein Bézier 类曲线和 Bézier 曲线的重新参数化方法. 计算机研究与发展, 2004, 41(6): 1016~1021
- 9 Jüttler B. A vegetarian approach to optimal parameterizations. Computer Aided Geometric Design, 1997, 14: 887~890