嵌入式RTOS安全保护机制的研究与实现

王丽杰，熊光泽，罗 蕾

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】在分析安全相关的ARINC653规范的基础上，提出了满足安全关键应用的嵌入式实时操作系统S-CRTOS体系结构，较详细阐述了该体系结构所采用的隔离和保护原理，以及所解决的区间调度、进程池和异步机制等几种关键技术。该体系结构已在多种目标板上成功实现，可用于航空电子、汽车电子等安全关键系统。

关 键 词 区间; 隔离和保护; 内存管理单元; 异步信号; 嵌入式实时操作系统

中图分类号 TP316; TP311 文献标识码 A

Research and Implementation of the Safety Protect Scheme for Embedded Real Time Operation System

WANG Li-jie，XIONG Guang-ze，LUO Lei

(School of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract Based on the analysis of the safety related ARINC653 specification, an scheme of embedded real time operating system being suitable for safety critical applications: S-CRTOS, is proposed in this paper. The principle of isolation and protection is introduced in details, and several critical technologies: partition schedule, process pool and asynonous mechanism, are also described. This design has been implemented at some target boards, and can be used for safety critical systems: avionics electronics and automobile electronics etc.

Key words partition; isolation and protection; memory management unit; asynonous signal; embedded real time operating system

在传统的嵌入式实时操作系统(Real Time Operating System，RTOS)中，内核和应用都运行在同一特权级，应用程序可以无限制地访问整个系统地址空间，导致应用的潜在危险动作可能会影响其他内核和应用的正确运行，甚至造成系统崩溃。这种情况在众多实时安全关键应用领域，包括服务关键的因特网设备、任务关键的国防与航空航天系统、生命关键的医疗设备，以及过程关键的工业测量与控制系统，是不可容忍的。因此，应考虑对内核与应用，以及各应用之间进行隔离保护。但隔离保护仅凭软件是无法完成的，需要微处理器支持内存管理单元(Memory Management Unit，MMU)和高级保护模式。国外各大嵌入式开发商已相继推出了具有高可靠性的内核和应用保护机制的操作系统。本文在研究了ARINC公司发布的ARINC653(航空电子应用软件标准接口)的基础上，借鉴其应用保护思路，提出了安全的嵌入式实时操作系统(S-CRTOS，Safety CRTOS)的设计思想，并在具有MMU和高级保护模式的目标板上实现。

1 ARINC653规范分析

ARINC653主要阐述模块化综合航空电子设备(Integrated Modular Avionics，IMA)使用的应用软件的基线

收稿日期：2004 – 11 – 02
基金项目：信息产业部电子发展基金资助项目
作者简介：王丽杰(1975 –)，女，硕士，助教，主要从事嵌入式实时操作系统和计算机网络方面的研究.

操作环境，定义航空应用与下层操作环境之间的接口和数据交换的模式以及服务的行为，并描述嵌入式航空电子软件的运行环境。

1.1　软件构成

航空电子核心模块软件包括应用软件和核心软件。应用软件是与核心模块功能相关的部分，根据应用的关键级别开发并验证。核心模块提供应用软件执行的环境，包括操作系统和一些系统函数(如内置测试和下载调试等特性)。位于应用软件和操作系统之间的应用执行(APlication EXecutive，APEX)接口定义了系统为应用软件提供的一个功能集合。利用该功能集合，应用软件可以控制系统的调度、通信和内部状态信息。APEX接口相当于为应用提供的一种高层语言。图1给出了核心模块软件各部分之间的关系。

1.2　分区和区间管理

分区(Partitioning)是ARINC653中的一个核心概念，是航空电子应用中的一种功能划分。分区的单位称为区间，区间内的每一执行单元称为进程。每一个区间具有自己独立的数据、上下文和运行环境，优点是能够防止一个区间的错误影响到其他区间，并可使整个系统容易验证和确认。区间包括空闲、冷启动、热启动和正常4种工作模式。每个区间所需资源在系统构建时指定，区间的创建在区间初始化时完成。操作系统启动应用区间时，区间进入正常运行模式。监测管理功能在响应致命错误时重启区间或停止区间的运行。如图2所示。

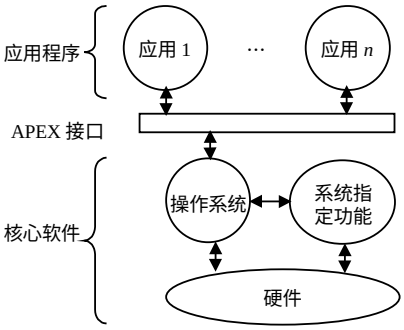

图1　核心模块软件结构

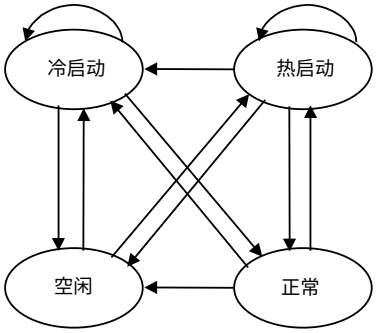

图2　区间状态转化模型

区间化以及区间的管理和调度是由操作系统来实现的。ARINC653为区间的调度规定了一种基于时间窗的循环调度算法。该调度算法的原理如图3所示。为了完成各区间的周期性调度，由操作系统维护一个固定时间长度的主时间框架，并在模块的运行期内周期性地重复。每个时间框架可以划分为若干个时间窗口。系统利用一个事先确定的配置表，在规定的时间窗口内激活对应区间的运行，就能够保证每个应用在分配给它的时间周期内访问公共资源时不被打断。区间在主时间框架内的执行满足：1) 一个区间可占有多个时间窗口；2) 允许有空闲的时间窗口，即不运行任何一个区间；3) 每个区间必须在规定的时间窗口内运行，使当前区间未处于运行状态时也不被其他区间占用。

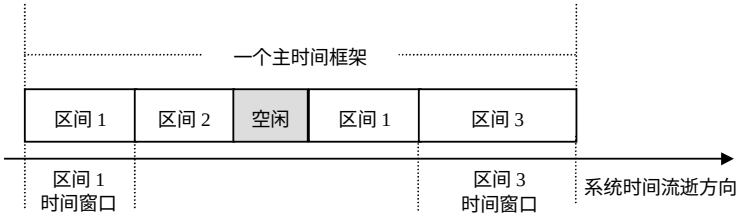

图3　基于时间窗的循环调度算法原理

2　S-CRTOS系统的设计与实现

S-CRTOS借鉴了ARINC653的区间保护和调度机制，其系统结构如图4所示。在该系统中，每个应用都工作在所属区间(Partition)的环境中，并且内核和应用以及各个应用之间都被保护墙(Protection Wall)隔离，无法相互破坏，保证了核心模块的可靠性。

S-CRTOS由操作系统核心层COS、操作系统区间层POS和模块支持层MSL等3部分构成。POS负责对区

间内的应用进程进行调度，并向上层应用提供事件、信号量等系统功能，其工作模式和职责与一般的多任务嵌入式实时操作系统类似。MSL实现硬件与COS的隔离，主要包括硬件初始化、上下文切换管理、CACHE服务、存储器分配服务、定时器管理、中断与异常管理等硬件相关功能。COS是S-CRTOS中最重要的部分，可分为基本核和可配置组件2个部分。基本核是操作系统运行所必需的功能模块，主要包括区间管理和调度、进程管理、内存管理、中断管理、出错处理、时钟管理、设备管理、异步信号和系统初始化；可配置组件给出系统中根据需要可进行动态配置的模块，由进程间的通讯机制、共享内存、文件系统等组成。

限于篇幅，以下只讨论COS中的隔离保护机制以及相关的几种关键技术的设计实现。

2.1 隔离和保护机制

隔离和保护是ARINC653主要强调的特性，也是安全的嵌入式实时操作系统必须解决的重点问题之一。S-CRTOS主要采用2种方式来实现应用与内核以及应用之间的隔离和保护。

第1种方式是使用MMU。MMU能够实现逻辑地址到物理地址的转化，并且对访问权限进行控制，既可保护系统内核不受应用软件有意或无意的破坏，也可有效防止各应用软件之间的相互破坏。图5给出了页目录/页表方式的MMU地址转换流程。

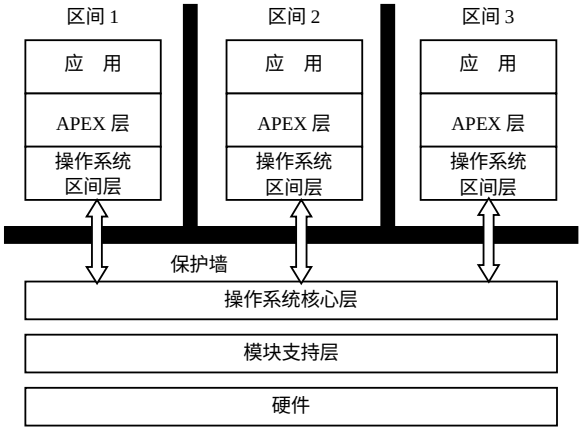

图4 S-CRTOS的系统结构

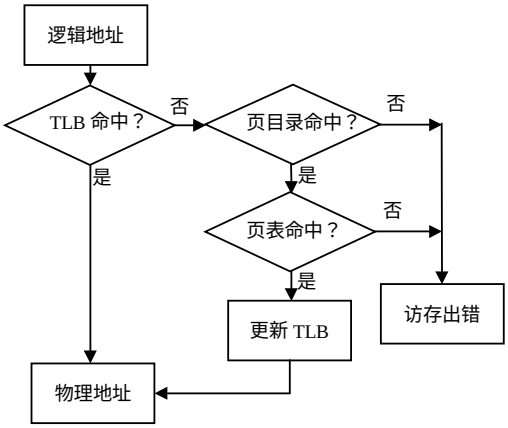

图5 页目录/页表方式的地址转换流程

第2种方式是TRAP(自陷)系统调用。S-CRTOS为实现对内核以及应用之间的保护，提供了用户态和系统态2种运行形态。操作系统内核运行在系统态，因此用户态的应用不能直接调用系统内核提供的功能接口，必须通过TRAP系统调用方式进行。用户态的应用需要调用内核提供的系统调用时，首先要执行一组特殊的指令使系统进入系统态，以执行需要的系统调用，调用完成后，内核将执行另一组特征指令将系统返回到用户态。每种支持保护模式的系统都提供了专门的软中断命令来完成从用户态进入系统态的功能(在x86中是int 0x80指令，MPC750中是sc指令)。系统挂接一个软中断处理函数，所有的系统调用都通过该软中断进入，并以不同的系统调用号来加以区分。

2.2 区间调度机制

ARINC653规定，区间调度模式的主要特征是：1) 调度单元是区间；2) 区间没有优先级；3) 调度算法是预先确定的，按照固定的周期重复，并且只能由系统集成者进行配置。每个循环中，至少要为区间分配一个区间窗口。

在S-CRTOS中，为了方便系统处理以及保持良好的调度特性，对ARINC653的规定进行了2项修订：1) 调度单元是区间和系统进程；2) 区间具有优先级。但这2项修订并不影响POS和应用程序使用者，所以在外部特性上仍然遵循ARINC653的规定。另外，S-CRTOS还引入了Kernel区间和Idle区间2个系统区间。Kernel区间的优先级最高，用于为整个系统的运行提供支持。一些系统级进程也属于Kernel区间，可以方便调度。Idle区间具有最低的优先级，用于填充系统时间。当系统中没有其他区间可以运行，就运行IDLE区间。

S-CRTOS区间的调度原则首先是基于优先级调度，对同一优先级的区间可使用时间片轮转调度或基于时间窗口的循环调度策略。S-CRTOS采用两级调度机制。系统具有区间的优先级位图和每个优先级对应的就绪链表。每个区间也包含系统进程的优先级位图和对应的就绪链表。利用优先级位图和就绪链表，可容易地实现对区间和系统进程的调度。图6给出了调度模型。

调度模块根据系统中的区间优先级位图和区间就绪链表进行区间调度，被调度的区间根据区间内的进程优先级位图和进程就绪链表进行系统进程调度。两级调度相对于一级调度，能够保证调度时间的确定性，系统调度时间不会因为区间和系统进程的多少而发生变化，符合实时操作系统的定义。在进行基于时间窗口的调度时，2个系统区间实际上也会参与调度。Kernel区间的运行时间会被计算在区间调度配置表中的当前区间的运行时间上，也就是说，对于某应用区间而言，它的运行时间实际上包含系统区间的运行。

2.3　进程池机制

在S-CRTOS中，进程池是指一组由系统进行维护的进程，这些进程能够为应用提供一组空闲进程的请求/回收服务。运用通过系统提供的调用接口对系统提出应用请求，系统把应用请求交给进程池，进程池自动选择一个空闲进程系统其进行服务，服务完毕以后，系统会回收该系统进程。

进程池主要应用于时钟定时器服务、中断服务和异步IO操作。传统的中断服务是中断之后直接执行中断服务程序(Interrupt Service Routine，ISR)，但S-CRTOS还支持中断服务处理以进程方式进行，系统可以再次响应低优先级和同级中断，不会造成中断丢失。此外，系统中断响应时间变快，有助于提高系统的响应能力和可靠性。

2.4　异步信号和异步IO机制

S-CRTOS支持异步信号机制，允许在某种情况下COS以异步方式发送信号到上层的POS。异步信号的一种典型使用是异步IO。在应用程序中，POS中的进程通过系统调用产生阻塞性IO请求时，若IO不是异步的，该进程就会阻塞，等待系统调用的返回。但是COS并不知道POS中的应用进程的存在，当其中的一个进程阻塞时，整个区间都会被阻塞。采用异步IO方式可避免阻塞情况的发生。

当应用请求了阻塞性IO服务时，S-CRTOS会启动一个专用系统进程来进行应用请求的IO操作，称为异步IO机制。S-CRTOS将为实现异步IO操作而创建的系统进程称为worker进程，如图7所示。在应用程序中，当POS中的一个进程调用异步IO系统调用时，该系统调用检查此IO是否为阻塞工作方式，如果是，则创建一个worker进程完成指定的IO工作，并返回一个AIO_PENDING值到POS。POS检查到这个返回值以后，把该进程从就绪队列取下放入等待队列进行重新调度。worker进程完成了要求的IO操作时，即发送异步信号到POS唤醒原阻塞进程。

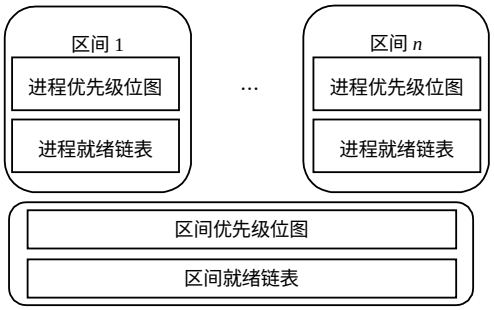

图6　S-CRTOS的区间调度模型

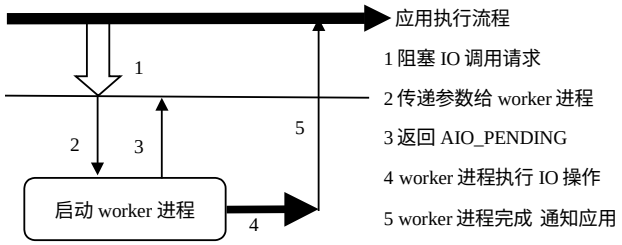

图7　异步IO工作流程示意图

3　结 束 语

基于ARINC653的应用保护思想设计的S-CRTOS能够满足航空航天、电子等安全关键系统对高可靠性和高可用性的要求。S-CRTOS系统方案已成功在多种目标平台实现，并逐渐被应用到航空、电子等相关领域。本文提出的设计思想有助于促进国内对嵌入式实时操作系统安全保护技术的研究。

参　考　文　献

[1] Aeronautical Radio, INC. Arinc Specification 653 Avionics Application Software Standard Interface[S]. 1997

[2] Aeronnautical Radio, INC. Supplement 1 to Arinc Specification 653 Avionics Application Software Standard Interface[S]. 2003

[3] 曾家智，王　蓉，刘乃琦. 80486/80386系统设计和应用[M]. 成都: 电子科技大学出版社, 1992

编　辑　熊思亮