安全(Safety)内核机制的研究与实现*)

The Research of Safety Kernel Mechanism in RTOS

黎忠文　熊光泽

(电子科技大学计算机科学与工程学院　成都610054)

Abstract In this paper, we discuss deeply safety kernel, which is the new concept of safety assurance, as well as difficult of developing safety-critical system. And in order to provide supporting platform for safety-critical system, an approach, which is to make use of safety kernel mechanism in RTOS, is presented. At last, we design a safety kernel mechanism for CRTOS Micro. It can address common problems found in RTOS supporting safety kernel, such as lack of the reuse and expressiveness of safety kernel, harmony between safety and real-time, etc.

Keywords Safety-critical system, Micro-kernel, Operating system, Safety kernel, Safety policy

1　引言

安全技术是控制系统,特别是安全关键系统必需研究的核心内容之一。随着各种计算环境的建立、应用需求的复杂化和系统结构设计的开放化,软件以其易获得、易修改等优势,正逐步成为控制系统的主角,用以处理系统内外的复杂关系,这是现代控制系统发展的显著趋势。然而,由于软、硬件故障机理的本质区别,现有的硬件及传统的软件安全可靠技术并不完全适合于处理软件故障,因此这种趋势增加了系统的安全隐患。比如,对于一般复杂度的软件,用测试的方法能使它的故障率降低到每小时10^{-4}个,相当于一年一个。但当软件复杂度增加时,测试效果急骤下降。若采用容错、检错等综合方法也只能使软件错误率降到每小时10^{-5}个,然而安全关键系统的故障率要求是每小时10^{-9}个,甚至有的是每小时10^{-10}个[1]。

对于任何一个细微故障都可能造成严重后果的安全关键系统而言,上述安全隐患是不容忽视的。事实上,安全工程师在设计系统时对软件的采用都比较慎重,在能使用硬件技术的地方尽量不用软件,特别是复杂软件,但对于现代安全关键系统不采用软件技术很难实现。为此,近年来世界各国纷纷加大了对安全软件的研究,主要集中在以下几个方面:

(1)安全软件工程的研究。对软件生命期各个阶段中,与安全相关的管理、开发、人员配置,技术监督和质验等进行广泛的研究。许多行业都已基本制定出相关的标准,比如美国军标 MIL-STD-882B 和 MCD 00-31、英军标 MOD 00-56、原子能工业的 IEC-880等。专用于安全软件开发的 CASE 工具,比如法国的 SCADE 工具、美国的 SpecTRM 工具,也相继推出。

(2)新的软件安全可靠技术的研究。主要根据软件故障的特点建立新一代软件安全可靠机理、形式化方法是其主要代表。

(3)研究新的安全保障机制,即新形势下如何防止事故的发生。安全内核(safety kernel)是新观点也是当前的主要研究方向之一。

(4)控制系统支撑平台的研究。目前该方面的研究工作比较少、难度也最大,主要涉及应用、操作系统等多种技术。当前的进展是设计专用于控制系统(主要针对非安全关键系统)的操作系统,比如 Chimera II。

我们以本校"九五"最新成果——基于第二代微内核思想的实时操作系统"CRTOS Micro"为依托,在全面分析 safety 内核思想的基础上,设计了基于实时操作系统的新的安全保障机制,以期为开发我国自主的安全关键系统提供良好的软、硬件支撑平台。

2　内核原理的概述

safety 内核的灵感来源于信息保密系统中较为成熟的安全内核,它最先由著名安全学家 Leveson[2]等人提出。但那时的 safety 内核与 security 内核相距甚远,更象是一个纯粹的监视器,没有安全保障及实施策略。真正从 security 内核视角看待 safety 的是 Rushby[3]。后来 King[4]继承和发扬了 Rushby 的 safety 内核概念。他对 MSS 和 UVAR 两系统的 safety 进行了研究,并在 MSS 上建立了 safety 内核的雏型。那么内核机制究竟在 security 和 safety 界扮演什么角色?下

*)本项目获"九五"国防技术预研项目资助。黎忠文　博士生,主要研究领域为多机系统的可靠性、安全性及 QoS。熊光泽　教授,博士导师,主要研究领域为实时计算机系统及其软件开发支持。

面将剖析之。

2.1 内核机制的动因

内核常用于实施分隔，security 内核的设计思想是把系统的 security 策略封装在系统的某组件(即 security 内核)中，以免被应用软件(即用户)及操作系统侵入，造成泄密、数据的非法修改和对系统合法访问的拒绝。只要应用软件对系统的访问与 security 相关，security 内核都必须根据 security 策略对它进行控制和处理，以确保整个系统的安全(security)。因而不需要系统其它组件的合作，security 内核就能独立承担维护系统安全(security)的重任，其原理[4]如图1所示。

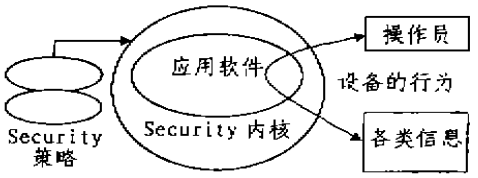

图1 security 内核原理

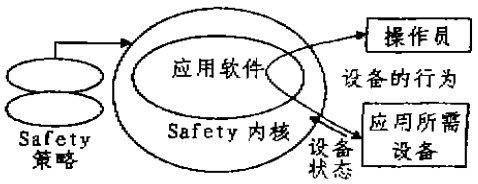

图2 safety 内核原理

虽然 safety 有别于 security，它关心的是如何控制应用设备，以避免重大的生命财产损失和环境的破坏，而不象 security 那样侧重于信息的保密性、完整性和可用性。但是 safety 内核与 security 内核的原理类似，如图2所示。凡是应用软件对系统设备的访问，都必须经过 safety 内核的审查控制，合法者予以支持，反之则采取相应的安全措施，维护系统安全(safety)。这样就能很好地避免软件错误造成的灾难后果。内核机制的优势[5]是：

·统一性：所有的安全策略都能由单一的代码集合来执行。

·可修改性：易于对安全机制作出改变和对这种改变进行测试。

·紧凑与可验证性：安全内核只完成安全功能，其结构相对较小，有利于正确性的验证，

·单纯性：把原本由应用软件考虑的安全工作交由内核处理，减少了应用软件的负担。

2.2 内核化系统与 safety 策略

内核机制非常诱人，但是不是所有的系统都利于内核化(采用内核结构)，所有的 safety 策略都能用内核来完成呢？

定义1(消极型错误) 当系统做了不希望做的事情时，称系统发生了消极型错误。比如，自动飞行控制软件在飞机处于飞行状态时，启动着陆装置。

定义2(积极型错误) 希望系统做的事情，系统没有完成时，称系统发生了积极型错误。

一般来说，积极型错误比消极型错误易于检查和评估，采用的方法也比较多，内核机制与大多数传统的 safety，可靠性方法不一样，它擅长处理消极型错误，事实上，无论 safety 内核外其它系统组件如何工作，safety 内核都必须要维持 safety 策略的不变性，换一个角度看，无论 safety 内核设计得多么好，它都没法保证系统其它组件是否正常调用了自己，相反，通过对内核的仔细设计，却能避免一些意外事件的发生。所以，诸如 safety 保障这类侧重于消极型错误处理的系统才可得益于内核化。

事实上根据前面对 safety 策略不变性的分析，我们知道只有一部份 safety 策略能在内核内实施，它们应满足的条件[4]可形式化表示为：

$$\forall\ \alpha \in op^{*} : P(op) \qquad (1)$$

其中 op^{*} 是内核提供的所有功能组成的集合，其元素(上式中以 α 代表)可以是内核的一个或多个功能；$P(\cdot)$ 则表示内核对 α 输入/输出的控制。事实上，内核为系统提供的每一次 safety 服务，都可看成是一系列有序的内核功能集(即 α)。因此(1)式说明，无论系统其它组件(特别是应用软件)行为如何(即 op^{*})，内核都要保持 safety 策略的正确实施(即 $P(op)$)。这类策略叫强安全策略，不适于用内核来实现的策略，即应用软件自行处理的策略叫弱安全策略。目前尚不存在 safety 策略的统一定义，不同行业、不同系统有不同的标准。但是对于一个具体的安全关键系统，比如核反应、自动飞行控制系统，safety 策略是具体的，它来源于系统的 safety 需求分析。一般来说，强安全策略分为四类：

(1)设备控制策略 当应用向某设备发出动作指令时，该策略用以判断该设备是否处于执行该动作的状态，是否会造成安全事故。如果可行，允许设备执行操作，否则出错处理。

(2)应用软件的状态策略 与硬件相似，在一个特殊的软件行为发生前，必须要检测这种行为发生的合理性，即严格控制软件的执行路径，避免错误指令的产生，对于每一个特殊的软件行为都要为之设计可接受的命令序列及参数，以备检测时使用。

(3)设备错误诊断策略 这类策略主要是检测设备的动作是否与指令动作一致。这类策略能处理软件之外的故障，比如设备出错、网络出错。

(4)错误响应策略 错误出现后，采取的补救措施。

强、弱安全策略的划分不是绝对的，设计者可根据自身的条件及易验证的角度进行考虑。

3 Safety 内核的外部结构

必须明确的是 security 内核与 safety 内核机制建立的出发点存在重要的区别：safety 内核假设系统中没有恶意的应用软件——它被设计为有意避开 safety 内核对系统进行破坏。而 security 内核则刚好相反，它是针对恶意软件设计的。因此，它们的实现方式完全不一样。目前 security 内核技术已经比较成熟，它往往与操作系统的内核设计为一体，比如著名的 security 操作系统 SCOMP 和 UCLA。而 safety 内核尚处于研究阶段，它在系统中的位置一般分为集中式和分散式两种[5]。

集中式结构中，safety 内核被直接置于硬件平台上，如图3所示。这种结构适用于应用软件比较简单，能在单机上完成的情况。但是对于复杂的应用软件，比如大型图像处理，单机很难满足其计算要求，这时就需把应用软件中与 safety 控制无关的部分放在它机上执行，于是形成了 safety 内核的第二种外部结构——分散式，如图4所示。分散式结构的特点是 safety 内核强烈地依赖于系统软件（如果不用系统软件，safety 内核就不得不提供类似远程管理的系统功能，这与简化 safety 内核的目标相抵触）的支持，当系统软件出错时，可能会导致 safety 内核的非正常工作。为此引入了被称为 safety 内核监视器的硬件机制，它监视 safety 内核和应用设备的状态，一旦发现 safety 内核出现问题，立刻让系统进入安全状态。

图 3 Safety 内核的集中式结构

图 4 Safety 内核的分散式结构

4 Safety 内核的实现

目前 safety 内核的实现[4,6]都属于手工作坊式，其设计管理完全由应用自行考虑，对系统软件而言，safety 内核纯粹就是应用软件。当设计每一个新系统时，safety 内核都必须由设计者从头做起，效率低、可靠性差，有碍安全关键系统的开发。于是我们提出扩展操作系统以支持 safety 内核，为开发我国自主的安全关键系统提供良好的软、硬件支撑平台。该方法面临的基本困难有：

(1)操作系统的选择。适于扩展后用于安全关键系统的操作系统必须是实时操作系统，而且要有较强的硬件抽象和支持系统移植的功能。

(2)safety 内核的表示。系统以什么样的抽象结构来描述 safety 内核，以达到记录不同应用种类的 safety 策略、从而重用 safety 内核机制、方便用户、利于验证和实现的目的。这是非常重要和关键的。

(3)safety 内核与操作系统之间的关系。safety 内核与操作系统之间的运作关系，如何协调安全性和实时性，使它们能同时得到保证。

4.1 CRTOS Micro

CRTOS Micro 目前 x86平面模式、分段保护模式和 DSP TMS320C3X 等三种实现版本，其结构[7]如图5所示。与传统基于微内核的操作系统构造不同的是，CRTOS Micro 不单采用 Client/Server 模式，同时采用可重入的共享库。此外，CRTOS Micro 通过纳核和板级支持包(BSP，Board Supporting Packages)提供强有力的硬件抽象和系统移植支持。其中，纳核用来支持微内核在不同处理器上的移植；BSP 则用来支持操作系统在各类 OEM 板或用户设计的专用硬件系统板上的移植，它与硬件直接打交道，并以设备库 DevLib 的形式提供给上层应用。

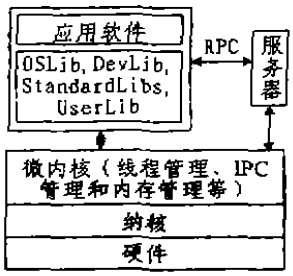

图5 CRTOS Micro 操作系统的结构

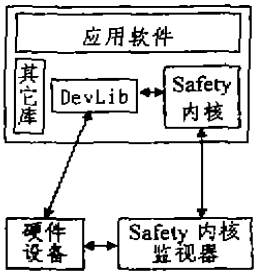

图6 Safety 内核的位置

4.2 Safety 内核机制在实时操作系统中的建立

利用 CRTOS Micro 支持可重入共享库的功能，把 safety 内核设计为用户空间的共享库，如图6所示。

safety 内核不以专用服务器的方式建立在系统中，于是当应用请求安全内核服务时可直接访问共享库，避免了远程通信的开销，保证了良好的实时性。另一方面，safety 内核在设备动作之前，通过 BSP 获得设备安全相关信息并实施 safety 策略，及时避免因应用软件、操作系统等系统软件及设备故障所造成的安全事故，保证了安全性。

safety 策略实际上可以看成是一系列的 safety 协议，我们用有限状态机(Petri 网等其它协议描述工具均可)来描述它，其中设备状态、应用软件状态为状态结点，设备动作指令与设备出错处理策略等为状态变迁的触发事件。映射器(见图7)设计为 safety 策略到有限状态机的自动转换工具。

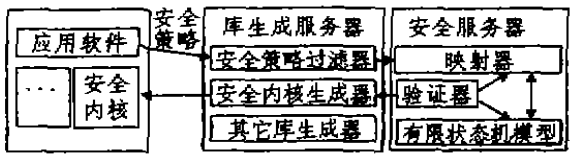

图7 Safety 内核的自动生成

Safety 内核自动生成过程采用专用服务器来完成(因为没有实时性要求,又可便于移植)。为此在 CRTOS Micro 中生成了 safety 服务器,在库生成服务器中添加了与安全相关的工具。主要过程如下(图7):应用通过 CRTOS Micro 的微消息通信机制把安全(safety)策略发送至库生成服务器;后者使用过滤器对安全(safety)策略进行分类(比如核反应类、交通控制类等),并把分好类的 safety 策略发送给 safety 服务器;safety 服务器中 safety 策略映射器根据当前系统设备的实际情况把 safety 策略映射到有限状态机上,由验证器对两者的一致性进行验证;验证完后的有限状态机在库生成服务器上生成安全内核库并放置在相应的应用空间,供应用软件使用。

结束语 safety 内核是当前最新的安全保障机制之一,本文在深入讨论 safety 内核的基础上结合安全关键系统发展的趋势,对 safety 内核与操作系统之间的关系、safety 策略的表示式及 safety 内核的建立等关键问题进行了研究,并在超微内核实时操作系统 CRTOS Micro 上设计了 safety 内核的实现机制。它具有安全性与实时性并举、safety 内核机制重用等功能,能为安全关键系统的开发提供有力的支撑。我们下一步将对其进行优化处理。

参 考 文 献

1 Browen J P, Stavridou V. Formal methods and software safety. IFAC Symposium, Zurich, Switzerland, 1992. 93～98

2 Leveson N G, Shumeall T J, et al. Design for Safe Software. In: Proc. AIAA Space Sciences Meeting. Reno, Nevada, 1983

3 Rushby J. Kernels for safety?. Safe And Secure Computing Systems Symposium. London: Blackwell Scientific Publications, 1989. 210～220

4 King R. Safety kernel enforcement of software safety policies: [Doctor Thesis] USA: University of Virginia, 1995

5 蒋继洪,黄月江编著.计算机系统、数据库系统和通信网络的安全与保密.成都:电子科技大学出版社,1995

6 Ammann P. A Safety Kernel For Traffic Light Control. 1995. Available at: http://www.isse.gmu.edu

7 王志平.硬实时操作系统研究:[博士论文].成都:电子科技大学,2000

(上接第66页)

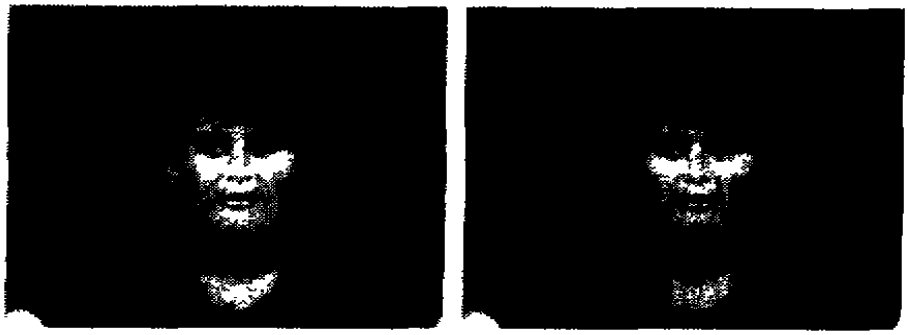

图7 Miss America 第26帧原图(左)和解码图像(右),CR=231.05,PSNR=37.88dB

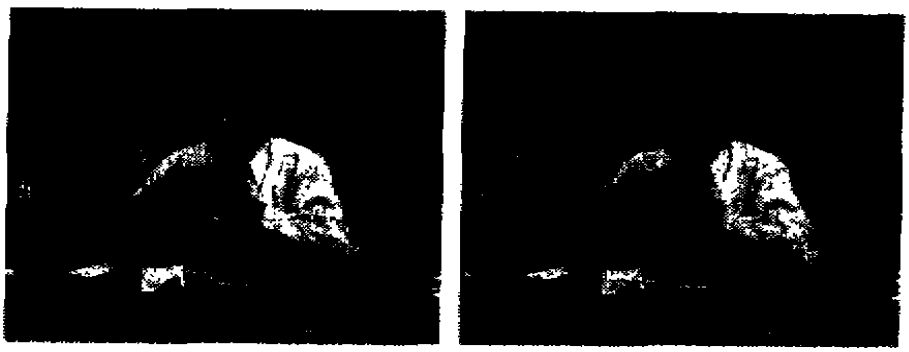

图8 Salesman 第16帧原图(左)和解码图像(右),CR=66.17,PSNR=32.38dB

2 Shapiro J M. Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Proc., 1993, 41(12): 3445～3462

3 Xiong Z, Guleguz O, Orchard M T. A DCT-based embedded image coder. IEEE Signal Proc. Letters, 1996, 3(11)