维普资讯 http://www.cqvip.com

计算机科学1999Vol.26№.11

实时操作系统接口标准规范——CTRON的分析

The Analysis on CTRON—an Interface Specification of RTOS

杨 早 张汝杰
（华东师范大学计算机科学系 上海200062）

Abstract Concentrating on the functions and structure of the CTRON, this paper analyzes its task management, task synchronization and memory management, and discusses its real-time characteristic and advantages.

Keywords CTRON, RTOS, Function, Structure

一、引言

在信息通信网络中存在信息节点、通信处理节点、交换处理节点、工作站节点等多种类型的节点。各节点功能有所不同，相应地提供各种信息通信服务。随着信息通信服务的多样化发展，针对网络的服务也相应增加。基于此种情况，要求把相同的功能装载于不同的节点上，并且节点间的功能划分也能轻易地改变。这就意味着节点间软件的流通性（再利用）成为一项重要工作。传统的OS也具有此类功能，但因为它们仅限于针对不同硬件种类或信息处理、通信处理等领域提供不同接口，所以并不能够很好地实现节点间软件流通。

为了实现节点间软件流通，要求各个节点上的操作系统提供一个统一、标准的程序接口。近年来，许多研究人员一直在为接口标准的制定而努力，ISO组织已经推出了可实施的POSIX.1标准，日本的TRON协会则推出了CTRON(Communicate Real-time Operating System Nucleus)作为实时操作系统的接口定义规范，主要用于通信领域，以保证通讯网络内部软件流通性、高可靠性和实时性。TRON协会成立于1988年3月，经过数十年的发展，已经被世界上很多软件生产厂商所重视。许多嵌入式实时操作系统的API接口都符合CTRON标准。如eCos系统等。文中分析所用到的接口规范和实现实例就是由TRON学会提供的。

CTRON主要针对网络应用，所以十分注重接口定义的实时特征和程序流程的稳定性与可靠性。分析CTRON的接口规范和实例，就是希望能够从中了解并掌握这些特点，有助于今后在实时操作系统方面的设计。

二、CTRON的功能分析

CTRON规定了一个实时多任务管理内核，主要提供了任务管理、任务间同步、内存管理这三方面的功能。

2.1 任务管理功能

在CTRON中可并行执行的最小程序单位被定义为任务。任务的管理功能有如下几种：

·任务生成、启动、结束、清除、强制结束和强制清除。

·任务睡眠、唤醒、挂起、挂起解除。

表1 CTRON中任务的状态列表

状态名称	解释
Non-Existent	任务不存在
Running	当前正在使用处理器
Ready	处理器以外的所有资源已准备好，只等待处理器的分配
Waiting	等待获得处理器以外某些资源（或等待事件的发生）
Suspended-Waiting	被其它任务从Waiting状态强制性地置于调度对象以外。这种状态被解除后，返回Waiting状态
Suspended-Ready	被其它任务从Ready状态强制性地置于调度对象以外，这种状态被解除后，返回Ready状态
Dormant	已确保提供了处理器以外的资源，但还不能作为调度对象

它所提供的任务管理API类似于UNIX的相应

维普资讯 http://www.cqvip.com

API，但CTRON中任务的状态数目较多，详细地划分了任务对CPU、其他资源、时间和运行权限的需求关系，使得任务的调度更加的灵活。如在通用操作系统中，一般不能挂起"就绪"和"运行"以外状态的任务，有时出于管理的目的，需要使一个处于"睡眠"状态的任务在被"唤醒"后仍不能执行，采用连续调用"睡眠"原语又会加重处理器的负担。在CTRON中，使用"挂起"原语就可将一个原处于"等待"状态的任务切换到"挂起-等待"状态。该任务等待的条件满足后，它不会切换到"就绪"状态而是切换到"挂起-就绪"状态，不参加任务调度。"休眠"状态标志那些已经"创建"但还没"启动"的任务，这些任务可以看作"备用"任务，减小系统中"请求任务"和"任务启动"的时间间隔，保证系统运行的稳定性。总的来说，CTRON中任务状态有如表1所示的七种，

CTRON中任务的转换流程如图1所示。

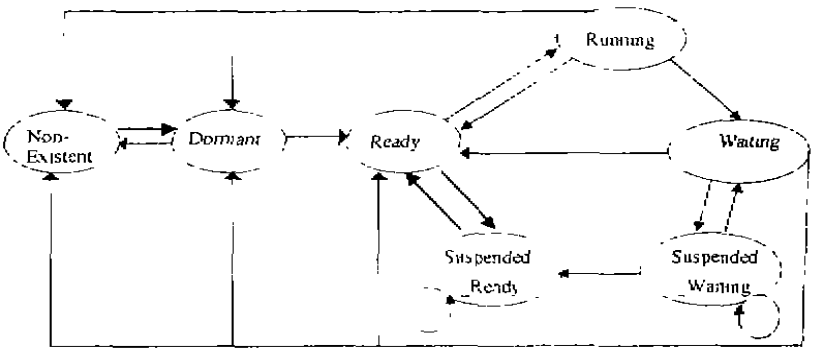

图1　任务状态转换图

表2　任务管理队列的定义与操作

队列名称	解释	使任务进入队列的API函数	使任务离开队列的API函数
Rdyque[]	就绪任务队列，队列按应用级别分成多级	任务启动Sta_tsk()，任务重启Rsm_tsk()，任务夭折Abo_tsk()	任务挂起Sus_tsk()，任务终止Ter_tsk()，任务退出Ext_tsk()
Tmque	有限时间睡眠的任务队列	任务睡眠sleep()	任务终止Ter_tsk()，任务唤醒wakeup()
Nulqueue_obj	资源不足或无权运行的任务队列(不参加调度)	任务创建Cre_tsk()，任务退出Ext_tsk()，任务挂起Sus_tsk()，任务终止Ter_tsk()	任务启动Sta_tsk()，任务重启Rsm_tsk()
Nulqueue_tm	无限等待的任务队列(不参调度)	任务睡眠sleep()	任务终止Ter_tsk()，任务唤醒wakeup()

在任务管理部分，分析的实例采用的多是链表的组织方式，体现了CTRON实时性的特点：

·对"就绪"(Ready)任务采用多级队列的方式进行管理，每一级队列对应一种应用级别，每一种应用级别中，再按任务的优先级进行排序。这样在调度任务时，可以给高应用级别的任务以更好的实时响应。

·如表2所示，对于任务调度中所使用的时间，等待、就绪等队列都采用了环形双向链的数据结构。这样使得由任意一个结点访问它的前趋与后继结点都十分的方便，提高了访问的速度。

2.2　任务间同步功能

CTRON构建的任务同步机构包括：事件(event)、邮箱(mailbox)和信号灯(semaphore)。这里着重介绍事件和邮箱，它们分别用于进程间标志性数据和少量数据的传送。任务在等待其它任务发来的事件和邮件的时候，都可以设定等待时限。事件或邮件超过时限还未到来，任务会被唤醒，这样可以检测任务间"死锁"情况的发生。

作为任务间最简单的同步机制，任一时刻事件只能被一个任务等待。和WindowsNT™相同，在CTRON中，事件标志由多个表示等待条件的标志位组成，可以单独对等待条件中的某一标志位进行设置。在判断事件标志是否满足等待队列中任务所需的要求

维普资讯 http://www.cqvip.com

时，有两个标准(由事件标志属性决定)：

·只要事件标志中有一个标志位满足等待条件，就满足了等待队列中任务的要求。

·事件标志中的各个标志位必须全部满足等待条件，才算满足了等待队列中任务的等待条件的要求。

在对事件标志进行设置时，针对原有标志位和新标志位之间的逻辑关系，可以对事件的每一个标志位单独的与新的标志位信息进行"与"、"或"、"置换"和"异或"操作。这样每个标志位就具有了"记忆"的功能，丰富了用户任务接收到事件的信息。

在事件标志满足了等待队列中任务的要求后，根据事件的属性，事件的各个标志位可能被复位，如果这时需要复位，说明这个事件只能被一个任务使用，这是一个同步性的事件；如果不复位，说明这个事件可以被多个任务使用，这是一个通知性事件，可以看出CTRON中事件的用法是很灵活的。

与一些操作系统不同，在CTRON中，发送邮件的任务不需等待接收方的回应(Reply)，发送邮件不会导致任务被阻塞。从图2来看，邮箱内部可容纳多个消息和多个任务，邮箱传递消息的方式有两种：一种为从送信方任务向邮箱传递，再从邮箱向收信方任务传递的接力(move)方式，另一种为在公共内存中设定消息、发送和接受消息位置指针的定位(locate)方式。两者的主要区别为，定位方式不需要传递消息的内容，而只要传递消息的地址，接力方式需要传递消息的内容。两种消息传送方式分别着重效率和可靠性方面。

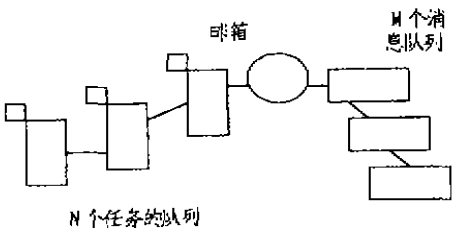

图2 邮箱结构示意图

CTRON以同一函数向邮箱发送消息，由邮箱根据自己的属性决定操作采用定位方式或接力方式。CTRON按不同的方式分为两个函数来从邮箱接受消息。对于要接受消息的任务，当邮箱中有消息时，可以得到一个消息的内容。当没有消息时，任务将被阻塞，被排在该消息的等待任务队列中。

2.3 内存管理功能

因为CTRON主要用于用在通信处理中，所以系统设计的重点为保证应用的实时性和稳定性。在由不同特性的应用程序混合起来的系统中，某一方的服务使用了过量的内存，就会妨碍其它服务的运行。例如，因为低优先级任务使用了过量的内存，高优先级的任务就变得不能执行，这种就是所谓的反向阻塞，而预先储存和分配用于特定用途的内存后，可以防止因内存不足而引起的处理中断，保证了处理执行稳定性。因此，CTRON内存管理的设计目标是：

·内存管理可以保证某些任务的内存分配。

·内存管理不会产生内存泄漏的情况。

·内存管理可以使任务不需要"忙等"内存分配。

·内存管理可以使任务检测出因内存分配导致的"死锁"情况。

·内存分配时，可以要求按申请任务的优先级排队。

·内存分配时，可以要求按申请的内存量的数目排队(要求量少优先)。

·内存块分配大小的可变性。

为了实现以上的设计目标，CTRON采用两级非耗尽式内存管理方式——池和块，引入了内存池控制功能。通过定义内存池，可以对应不同目的设定负载量并使用。例如，可以为处理特定通信路径的缓冲器定义内存池并进行分配。因此，内存池可以看作预先储备的用于特定用途的内存。

任务向内存管理模块申请内存时，是向某一个内存池申请一个内存块。如无可用的内存块时，它可以立刻以失败返回，也可以进入该内存池的等待队列，根据任务提供的计数值，可以是限时等待或是无限等待，这样任务在申请内存时不必多次重复请求，并可以检测出因内存分配导致的"死锁"情况，内存块队列、内存池和等待任务队列之间的关系如图3。

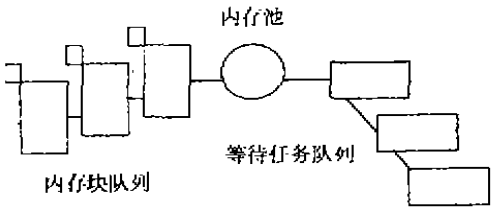

图3 内存池的结构示意图

在内存池生成的时候，可以设定它的属性，包括：

·内存池中每一个内存块是否是定长的？

·内存池中是否设立等待队列？

·内存池的等待队列的排序原则(FIFO，任务的优先级，申请内存量等)。

在任务申请内存时，内存池根据自己的属性采用

维普资讯 http://www.cqvip.com

不同的分配策略。

在系统启动时，CTRON 的内存管理模块将其可管理的最大范围内的内存转化为内存池 pool0，作为它管理的对象，所有任务所需的静态和动态分配内存都从这里分配。在 pool0中再划出若干个属性不一的内存池，供不同用途的任务使用，这些内存池中又分为若干内存块。这些可用内存块用双向链表构成队列，头指针放在内存池控制块中。在池一级存放等待分配内存的任务的信息。

在实现中，为了提高内存的利用率，减少释放-分配之间的时间间隔，CTRON 的内存管理模块处理释放的内存块时，可以不重新对可用内存块按大小进行排序，分配内存时，可以不采用"寻找最适合内存块"的策略，而采用"寻找最先适合内存块"的策略。

三、CTRON 的结构分析

在CTRON 规范下开发的多任务管理内核符合模块化、层次化的要求。从图4上看，CTRON 规定的各功能模块结构非常清晰，各模块并不直接使用其它模块向用户任务提供的 API，而是调用 Inner 模块所提供的内部 API。例如任务管理模块和同步管理模块会利用 Inner 模块的 Kr_free()和 Kr_malloc()函数来间接调用内存管理模块的功能。Inner 模块实现对下层硬件平台的虚拟化，使上层模块实现硬件无关性，主要针对中断管理模块和内存管理模块，例如 CTRON 提供了与硬件无关的16级中断管理机制。模块化、层次化的结构对代码的维护和修改都有益处。当改变下层硬件平台时，只要修改相应的 Inner 模块就可以，其它模块无需改变。

结束语 在多任务实时操作系统的环境下开发实时应用系统可以起到事半功倍的效果，不仅可以大大加快开发速度，而且使整个系统的稳定可靠性、可维护性大大提高。以上对 CTRON 内核的功能和结构两方面的分析，说明了 CTRON 接口规范具有实时性、结构的完整性等优点，可以作为今后多任务实时操作系统的一个开发基础。

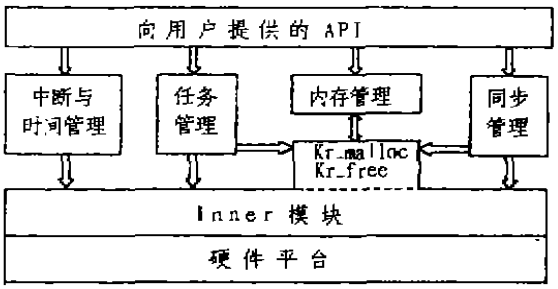

图4 CTRON 的结构示意图

同时，由于 OS 接口的共同化，即使计算机厂商不同，机种（大型、小型、工作站等）不同，用户的应用程序或软件公司销售的捆绑程序（源代码形式）均可不用进行修改。用户及软件公司就可削减软件开发的费用及缩短开发周期。因此，CTRON 规范的提出是有很大的发展前景的。

参 考 文 献

1 和佐野哲南，小林吉纯、面向信息通信网络的 OS 接口—CTRON. 1989

2 耿技，周明天. 现代操作系统概念. 计算机科学，1999，26（3）：19～23

3 Ford B. Interface and Execution Models in the Fluke Kernel. Operating System Review. 1998

（上接第52页）

站点，在 Web 页上找到该书，然后，以 Neuron 来运行。

使用 Plug-in 运行 ToolBook Ⅱ书也存在着以下两种缺陷：ToolBook Ⅱ菜单将不能被显示；不论在安全模式还是在非安全模式中运行，均不能保存 ToolBook Ⅱ书中的任何变化。

小结 在网络化教学蓬勃发展的今天，急需开发一大批优秀的基于 Internet 或 Intranet 的 CAI 软件，然而现有的传统 CAI 软件是多年来广大教育工作者和计算机开发人员积累的宝贵财富，若能够在此基础上进行二次开发，将大大提高网络化教学的发展进程。

参 考 文 献

1 纳塔尔. D. Asymetrix Multimedia ToolBook 4使用指南. 北京：宇航出版社，1998

2 A Guide to Creating Interactive Courses about ToolBook Ⅱ Instructor，1996

3 Grand M. Java Language Reference. USA：O'Reill & Associates，1997

4 张卫民. Java 语言与 WWW. 北京：人民邮电出版社，1997

5 Sankar K. Internet Explorer Plug-In and Activex Companion. USA：Que Corporation，1997