

Vxworks 下主控并行处理分析*

朱辉庆

(海军驻杭州地区军事代表室 杭州 310012)

摘要 实时操作系统下的并行处理是许多较为复杂实时处理中不可缺少的一种处理方法。文章从并行进程的特点进行分析,结合 Vxworks 具体操作系统对有关事例进行浅析,提出部分实际情况下一一些编程方法。

关键词 并行处理;任务;操作系统

中图分类号 TP311.1

Analysis of the Parallel Processing Based on Vxworks

ZHU Huiqing

(Military Representative Office of Navy in Hangzhou, Hangzhou 310012)

Abstract The parallel processing in real-time operating system is an indispensable method in many of the more complex real time processing. From the analysis of characteristics of the parallel processes and relevant examples based on the Vxworks operating system, this paper proposes the programming methods in the actual case.

Key Words parallel processing, task, operating system

Class Number TP311.1

1 并行处理特点

并行^[1]是一种自然现象。在现实世界中,任何时候都会有许多事件同时发生。当设计软件以监测和控制现实世界中的系统时,就必须处理这种自然的并行。此外并行处理可提高处理速度、提高系统的可控性。但当处理软件系统中的并行问题时,常常有两个非常重要的方面:能够检测并响应以任意顺序出现的外部事件;确保在要求的最短时间内作出响应。

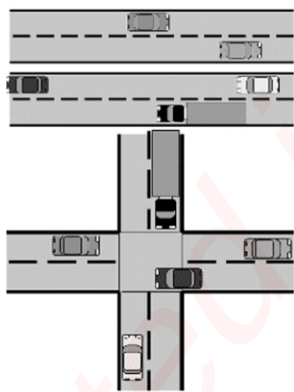


图1 公路交通系统

公路交通系统可作为一种形象的类比。不同道路上的平行车流基本上不会相互影响,所以很少产生问题。相邻车道上的平行车流为了交互时的安全,需要作一些协调。而在十字路口则会发生非常剧烈的交互,在这里需要进行周密的协调。因此,许多

实时软件系统都必须“反应灵敏”,而且可能在任意时间以任意顺序产生、执行。

但并行软件它有许多困难,大多数计算机和编程语言从发明之日起就属于顺序设计。一个过程或处理器在某一时刻只执行一条指令。而很多中小系统只有一个处理器,在单个顺序处理器中,必须通过交替执行不同任务来造成并行的假象。困难并不在其技术细节上,而在于决定何时和如何交替执行可能交互的程序段。

对于多个处理器,并行虽然很容易实现,但是交互情况却变得更加复杂。首先,运行于不同处理器上的各项任务之间的通信就存在问题。由于通常要涉及到软件的若干层次,因而会增加复杂性和定时开支。由于时钟和定时有可能不同,而各构件又可能会分别发生故障,所以多 CPU 系统的确定性也会有所降低。

要支持并行,系统必须提供多个控制线程。控制线程的抽象可由软件和硬件通过多种方式来实施。最常用的机制为以下某种机制的变形即多处理、多任务、基于应用程序的解决方案。当操作系统提供多任务时,常见的并行单元为进程。各进程可能有三种状态:1) 阻塞 等待接收某一输入或获得某一资源的控制权;2) 就绪 等待操作系统安排它执行;3) 运行 实际使用 CPU。

此外,通常会给进程分配相对的优先级。操作系统内核根据进程的状态、优先级以及某种调度策略来确定在任一给定时刻应当运行哪一个进程。实际上,多任务操作系统在其所有进程中共享一个控制线程,争用共享资源

2 Vxworks 下多任务共享数据并行控制与缓存

Vxworks^[2~8]是一个运行在目标机上的高性能、可裁减的嵌入式实时操作系统，Vxworks 实时内核—Wind 提供了基本的多任务环境^[9~10,13]。

多任务在单个处理器上构造出多线程并发执行的假象，实际上，系统内核是根据某个调度算法交替执行的。每个任务拥有各自的上下文，即拥有各自的 CPU 环境和系统资源，并保存在任务控制块(TCB)中。

任务的上下文包括：

- 任务的执行点(任务的程序计数器)；
- CPU 中的寄存器和浮点寄存器；
- 动态变量和函数调用所需的堆栈；
- I/O 操作分配的标准输入、输出和标准错误输出操作；
- 一个延时定时器；
- 一个时间片定时器；
- 内核控制结构；
- 信号句柄；
- 用于调试和性能监视的值。

与其他操作系统一样，任务执行是首先必须建立任务，激活任务。在 Wind 内核中，默认算法是基于优先级的抢占式调度算法，但在优先级相同情况下，可使用轮转调度算法，任务间一般用访问共享数据结构、消息队列等，在 Vxworks 中信号量被高度优化，并提供了最快的任务间通信机制。信号量是互斥和同步的最主要手段。在 Vxworks 实时处理系统中各任务间互斥和同步是至关重要的，以一个发射机发射与综合处理机状态查询两个任务为例进行说明。

具体工作过程为发射机接收综合处理机发射命令，发射一定时间信号，同时另一个任务综合处理机以一定周期进行状态查询，如两个任务工作正常时时序见图 2。

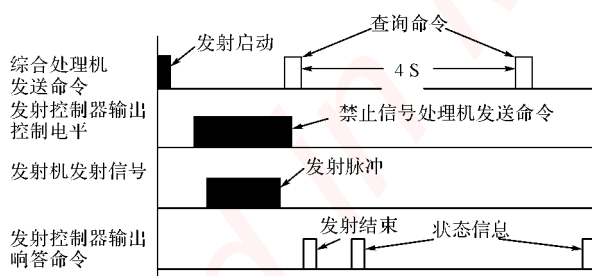


图 2 工作时序图

但由于综合处理机控制软件不一定同步，有可能使“状态查询命令”的实时发射时间落在“禁止综合处理机发送命令”期间，则造成的结果是综合处理机控制已向接口（综合处理机内一个接口管理 CPU）发送了“状态查询命令”，但接口控制模块由于发射机的禁止发送而将命令丢弃，从而使发了“状态查询命令”，但收不到发射机的“状态信息”并处于等待状态，一旦超过设定时间限制，综合处理机会判发射机故障，造成虚报。为此若增加一个同步、互斥量就可避免此现象，即在定时查询时在一个周期内将发射启动至发射结束这一期间提出，并在两个任务之间加一个握手，在发射任务内设置一个信息表示正在发射，排斥其他任务查询

发射状态，等到该信息量清空时，其他时间内仍可定期查询，当然，除此方法外，还有其他办法，在此举例只是为了说明 Vxworks 下同步与互斥的重要性。

在 Vxworks 中实时大流量命令控制及处理任务并行，尤其是许多设备响应时间不一时，共享数据的栈缓存的控制也很重要，以综合处理机的命令接收、命令处理两个并发任务为例进行说明。其中：

- 命令接收任务完成操控台不断发来的命令，并缓存进栈；
- 命令处理任务完成从共享栈中取出命令，并进行命令分类、响应；
- 状态查询任务完成中央控制件对外各设备状态查询。

设栈长度为 64，则综合处理机用接收任务接收命令并将 IP 指针加 1，命令处理任务读走一条命令响应，IP 减 1，但由于两个任务公用一个栈，而且任务间处理速度相差较大，栈的管理算法很有讲究，对栈的操作同样有一个同步与互斥问题，在实际实时系统中往往由于同步与互斥没处理好，使得命令处理任务读出了无效命令即部分命令参数被命令接收任务所修改，在对栈空与满处理上也很容易发生误判，导致实时系统工作效率低，易造成接口间死机。

此外，在 Vxworks 下许多实时信号处理系统中处理机一般都通过 VME 总线对多块 DSP 板共享内存进行访问，如图 3 所示。

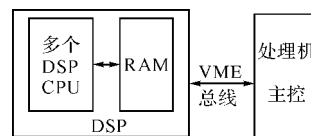


图 3 DSP 与主控数据交换示意图

如果 RAM 中数据格式示意如图 4：

RAM0	工作开始标志
0:	表示工作无效，所有数据造旧
1:	表示工作有效，显示新数据
RAM1	DSP 传送数据标志
0:	无效，DSP 正在写，主控禁读
1:	有效，DSP 已写完，主控可读
RAM2	显示画面数据
RAM3	显示画面数据
.....	
RAM(N)	显示画面数据

图 4 RAM 中数据格式示意图

从图 4 中可以发现，处理机的主控与 DSP 板之间的数据交换最好采用软握手同步方式，由于主控是一个多任务系统，DSP 板也是多 CPU 的处理，工作流程正常情况下按常规 DSP 写数，主控然后去读数一般是不会有问题的，这可以通过信号处理算法及硬件速度进行设定，但当主控流程或 DSP 处理流程受其他接口、任务干扰后，DSP 与主控之间的数据交换时序易“错位”，从而使主控读不到传送数据有效标志，也就不会去读 RAM 上画面显示数据，如果在 DSP 和主控之间增加一个软握手信号，即 DSP 在数据写入时增加判断主控标志是否为 0，为 0 则可写入，为 1 时就进行等待，主控读出时判断 DSP 标志是否为 0，0 时表示数据

(下转第 108 页)

表明上述评比的一致性很好。

求各行矩阵乘积的 $1/m-1$ 次方数值：

$$W_1 = \left(1 \times 1 \times \frac{1}{3} \times 2\right)^{\frac{1}{3}} = 0.874 \quad (30)$$

同样 $W_2 = 2.620, W_3 = 0.5, W_4 = 0.874$ 。

归一化得到：

$$\begin{cases} \delta_1 = \frac{W_1}{W_1+W_2+W_3+W_4} = 0.180 \\ \delta_2 = 0.538, \delta_3 = 0.102, \delta_4 = 0.180 \end{cases} \quad (31)$$

依照上面的办法可以求得指标的权重向量集合。

层次分析法的最终模型为

$$\begin{aligned} A_{t+T} = & \delta_1 x_{t+T} + \delta_2 (\omega_1 x_{t+T} + \omega_2 Q_{t+T} + \omega_3 y_{t+T}) \\ & + \delta_3 u_{t+T} + \delta_4 S_{t+T} \end{aligned} \quad (32)$$

对于不同的雷达,不同的使用情况,权重系数 $\delta_1, \delta_2, \delta_3, \delta_4$ 量都会相应变化,这里对于该型雷达,根据式(31)的计算结果: $\delta_1 = 0.180, \delta_2 = 0.538, \delta_3 = 0.102, \delta_4 = 0.180$, $x_{\text{单装}}$ 为单装使用数, $x_{\text{重要度}}$ 为前文所述的器材寿命数列, T 的起始时间从 2009 年算起,最终得到模型:

$$\begin{aligned} A(T, x_{\text{重要度}}, x_{\text{单装}}) = & 0.180(2.180 + 1.750T) \\ & + 0.217(2.346 + 1.50T) \\ & + 0.163(1.992 + 2.820T + 0.452T^2) \\ & + 5.517(1 - e^{-0.126})e^{0.126T} \\ & + 0.102(1.044e^{0.502x_{\text{重要度}}}) \\ & + 0.180(0.204x_{\text{单装}} + 0.571) \end{aligned} \quad (33)$$

通过计算可得,层次分析法的预测值为 4.83,这里与实际消耗量 4 比较接近。

4 结语

该模型对研究基层级雷达装备维修器材消耗的规律有一定实际意义,为预测器材消耗量提供了一种较为有效的预测模型。预测模型的建立对基层级器材的保障提供了依据,为雷达装备维修器材管理和决策人员提供了更加科学、准确的辅助决策信息,具有重要的实用价值。

参考文献

- [1] 单志伟. 装备综合保障工程[M]. 北京:国防工业出版社,2007.
- [2] 申宏伟,黄树军,万新敏,等. 一种雷达群组网系统战时备件需求预测方法[J]. 空军雷达学院学报,2010,24(1):31-33.
- [3] 黄勇,邢国平,孙德翔,等. 军机备件消耗影响因子的灰色关联分析[J]. 飞机设计,2011,31(4):74-75.
- [4] 罗轶,吴承广. 加强新型导弹装备器材消耗规律研究[J]. 海军航空工程学院学报,2000,15(4):489-491.
- [5] 陆飞龙,黄永海. 影响军械维修器材需求的因素及管理要求[J]. 装备维修,2008(10):36.
- [6] 王振龙,胡永宏. 应用时间序列分析[M]. 北京:科学出版社,2007.
- [7] 肖新平,宋中氏,李峰. 灰技术基础及其应用[M]. 北京:科学出版社,2007.
- [8] 王炯琦,周海银,吴翎,等. 基于模型概率的多模型融合定轨建模及仿真[J]. 系统仿真学报,2008,20(17):4722-4726.
- [9] 袁志发,宋世德. 多元统计分析[M]. 北京:科学出版社,2009.
- [10] 王连芬,许树柏. 层次分析法引论[M]. 北京:中国人民大学出版社,1990.

(上接第 90 页)

可从 RAM 读出。

3 结语

Vxworks 是目前较流行的一种嵌入式操作系统,在此环境下进行多任务系统开发,不但有一般多任务系统的共性,还有其特殊性,本文用工作中实际例子加以说明任务间同步和共享管理,起一个抛砖引玉作用。

参考文献

- [1] 陈国良. 并行计算结构、算法、编程[M]. 北京:高等教育出版社,1999.
- [2] 李方敏. Vxworks 高级程序设计[M]. 北京:清华大学出版社,2004.
- [3] 罗国庆. Vxworks 与嵌入式软件开发[M]. 北京:机械工业出版社,2003.
- [4] 陈育智,温彦军,陈琪. Vxworks 程序开发实践[M]. 北京:人民邮电出版社,2004.

- [5] 李方敏. Vxworks 高级程序设计[M]. 北京:清华大学出版社,2004.
- [6] 周启平,张扬. Vxworks 程序员速查手册[M]. 北京:机械工业出版社,2005.
- [7] WindRiver. Vxworks Kernel Programmer's Guide. 6. 8. USA: WindRiver Systems, Inc,2009.
- [8] WindRiver. Vxworks Application Programmer's Guide. 6. 8. USA: WindRiver Systems, Inc,2009.
- [9] WindRiver. Vxworks WorkBench User's Guide. 3. 2. USA: WindRiver Systems, Inc,2009.
- [10] WindRiver. Vxworks WorkBench Host Shell User's Guide. 3. 2. USA: WindRiver Systems, Inc. 2009.
- [11] 唐力强,王行祥. 嵌入式 VxWorks 系统下的 CPCI 设备驱动程序开发[J]. 计算机与数字工程,2011,39(1).
- [12] 胡俊. 基于 VxWorks 操作系统的 WindML 图形操控界面实现方法[J]. 计算机与数字工程,2011,39(11).
- [13] 孔祥营,柏桂枝. 嵌入式实时操作系统 Vxworks 及其开发环境 Tornado[M]. 北京:中国电力出版社,2002.