

一个分布式实时操作系统的设计与实现

谷建华 杨志义

(西北工业大学, 西安, 710072)

摘要: 本文论述了一个面向航空电子综合化系统的分布式实时操作系统的设计与实现。介绍了该系统的总体结构、系统软件中的任务管理、信箱管理、中断管理、I/O管理和分布通信管理等功能。

关键词: 航空电子综合化系统, 分布式实时操作系统, 分布通信。

一、引言

分布式计算机系统以其可靠性高, 容易扩展, 与应用环境相匹配等优点, 在比较复杂的实时控制系统中得到了广泛的应用。我们开发的“1553 B 总线通信实验系统”是一个分布式计算机系统。它由 8 台 PC/XT 286 构成一个两层总线系统。每个结点机由 MBI 接口板与 MIL-STD-1553 B 双余度总线互联。其中 MBI 板是作为 PC/XT 286 微机的一个插件板工作的。总线体系结构如图 1 所示。其中结点 A、B、C、D 构成上总线系统, 结点 C、E、F、G、H 构成下总线系统。

结点 C 上有两块 MBI, 分别属于上、下总线系统。它即是上总线系统的 RT (Remote Terminal), 又是下总线系统的 BC (Bus Control)。上总线系统各结点的数据要通过它才能传给下总线系统的各结点, 反之亦然。

本系统是面向航空电子综合化系统的分布式实时操作系统, 它不仅具有一般分布式实时操作系统的共同特性, 还体现了航空电子综合化系统特有的阶段控制性、处理复杂性、任务确定性、系统容错性等特点。在设计时, 我们考虑了上述特征, 使系统更具特色。本系统软件驻留在每个结点机上, 负责本结点上资源管理, 为应用任务提供实时服务, 实现各结点机的通信, 将物理上分散的多台计算机联合成一个整体。系统软件包括核心执行软件和分布执行软件两大部分。

二、核心执行

核心执行软件完成多道任务管理, 任务之间的同步, 以及时钟和 I/O 管理等。

1. 任务管理

考虑系统的实时性和功能分布的确定性等特点, 该系统采用静态任务分配, 运行中无任务的动态迁移。这就省去了在系统运行过程中动态生成任务的许多开销, 提高了实时性。

在不同的时刻, 任务处于不同的状态。各种状态在不同的条件下可相互转换。考虑本系统任务运行的阶段性特点, 我们将任务分为激活与非激活两种状态。并约定, 只有处于激活状态的任务才能参与调度。当系统的工作模式发生变化后, 可让在该模式下不运行的任务处于非激活状态, 而需要运行的任务处于激活状态。这就可使在某一阶段内暂时“无用”的任务不参与调度, 从而减

本文于 1991 年 12 月收到。谷建华, 1965 年生, 1987 年毕业于西安电子科技大学计算机系, 1990 年在该校获硕士学位, 主要从事“分布式实时计算机控制系统”的研究。

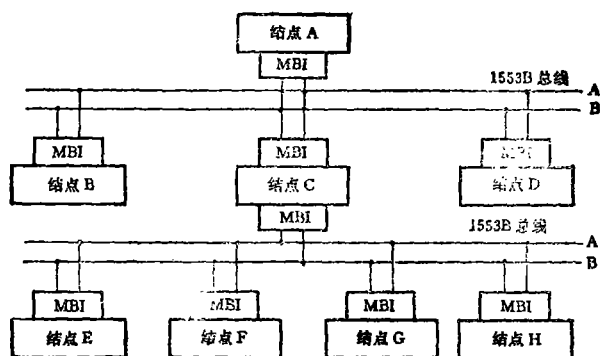


图 1 系统结构框图

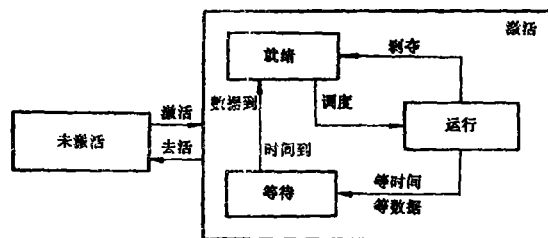


图 2 任务状态转移图

少了开销，提高了实时性。

处于激活状态的任务又进一步细分为就绪、运行和等待三种状态。它们的含义分别如下，

就绪：任务具备运行条件，排队等待处理机资源，一个任务被激活后首先进入就绪状态。

运行：任务正占有处理机资源。在任一结点中，处于运行状态的任务只能有一个。

等待：任务因某种条件不满足而暂停等待，一旦条件满足，任务便从等待状态变为就绪状态。

任务的各种状态及转换如图 2 所示。

处理机调度分为两级：任务分配和任务调度。前者是指从全系统角度出发，对各结点合理分配负荷。该系统对任务的分配采用事先分配的原则，即在系统分配之前就根据系统的特点，确定各个结点上的任务数和具体任务，在系统运行过程中不再改变。后者则指在某个结点范围内对处理机的调度。

基于优先级的剥夺式调度策略是实时操作系统的基本调度策略。其特点是实时性强，能充分体现任务间的轻重缓急。本系统采用剥夺式静态优先级策略，对优先级相同的任务，则采取先来先服务的策略。任务剥夺的时机当且仅当系统中有任务状态发生变化时。

每个任务都有一个任务控制表 TCB，它包含任务名、任务优先级、任务当前的状态、任务工作区及与任务有关的队列指针等。各个任务按其状态构成不同的 TCB 队列。当任务因不同的原因处于不同的状态时，任务管理软件就将该任务放到不同的队列中。系统根据需要，对相应队列实施操作。按队列操作的优点在于查找速度快、效率高。

2. 信箱管理

为了支持每台计算机中多个任务的实时并发运行，核心执行软件必须提供任务之间进行通信的手段，以实现任务之间的同步。该系统采用信箱方式实现。

信箱与具体任务是一一对应的，即每一个任务身上都背有一个信箱。给某一信箱发一个数据，就相当于给一个特定的任务发数据。信箱在系统生成时随任务同时生成。这样就减少了任务动态申请信箱和系统管理信箱系统的开销。

信箱总是属于信件的接收者。一个任务只能在自己的信箱上接收信件(数据)。但一个任务可以向除自己以外的其它任务的信箱上发送信件。

每个信箱又分为若干个信格，每个信格存放一个信件。每个信件有一个信件标识和一个信件属性。信件标识用于区分同一信箱上的不同信件，信件属性指明信件是否“挂号”，即指明发该信件的任务是否要等到接收方已收到该信件后，方可继续执行。在每一个信格中不存放实际的信件内容，只存放信件的存储区首址和数据长度。这样可使信箱的长度、格式统一，减少数据的搬动

次数。

一个任务要向另一个任务发送数据时,要动态地申请一块存储区,把数据写入;当另外一个任务收到这个信件后,把该存储区释放掉。当一个任务从信箱上读信件时,如果该信件不存在,则可以根据情况,或挂起该任务,等所需信件到来后再唤醒该任务,或直接返回该任务。当一个任务向另一个任务发送信件时,如果该信件是“挂号”信,则需要收方任务取走该信件后,该任务才能继续运行;如果该信件是“平信”,则不需要等到收方取走信件。为了提高实时性,在发送信件时,如果发现已经有任务正在等待该信件,则只需把该信件直接传给等待任务,而无需再开辟缓冲区,把信件写入信箱。每一个信箱上所占有的信格数是一定的。当一个任务向另外一个任务发送信件时,如果收方信箱的信格已经全部占用,则也需要挂起发送任务,等有空信格时再解挂它。

3. 中断管理

本系统的中断源主要有时钟中断和MBI中断。为使程序简单清晰,系统可靠实用,在中断处理期间不允许响应新的中断。为了避免在中断程序中逗留过长的时间而影响系统的实时性,在设计上采用了中断处理和中断任务相结合的原则,即中断处理程序只作些简单而关键的操作,然后立即从中断处理程序中返回,其余的工作留给有关任务去处理。在这些任务运行过程中可以响应新的中断,从而不影响中断响应的实时性。

(1) 时钟中断:系统中每个结点都有自己的时钟为实时任务提供支持。这里我们直接使用了PC/XT 286上的时钟发生器。设定每过一特定的时间,8253向处理机发一次中断。每次中断,标志一个解算周期的开始,中断处理程序首先要将该结点的系统时钟加1,然后将等待在时间队列上且等待时间已到任务变成就绪状态,完成对任务的时间驱动。为了使整个系统有一个统一的时钟,每过一定的时间间隔,主结点(AV)要向它的所有从结点以广播方式发一个时钟校正信息,所有从结点以此为依据,校准该结点的系统时钟。

(2) MBI中断:MBI的中断处理与分布通信密切相关,有关部分将在分布执行部分介绍。另外MBI上的一些错误处理,例如:MBI自检出错、在通信过程中的一些错误、主机向MBI发送了错误的控制码等。MBI也要向主机发中断,谋求主机对这些情况进行处理。

4. I/O 管理

本系统是在PC/XT 286机上实现的。整个系统的软件经编译,连接后,一次投入运行。根据系统的特点,我们直接延用了MS-DOS的I/O,文件和内存管理等功能。这样大大地缩短了系统的开发周期。但由于MS-DOS是面向个人的计算机,其最大缺点是许多模块不可再入。如果不很好地处理这些功能,很容易引起“死机”。所以,我们把一些有关I/O的操作(例如屏幕输出等)作了改写,从而避免了这些问题。

三、分 布 执 行

分布执行软件实际上是系统的通信软件。实现各结点机之间的通信。根据本系统的特点,系统的通信共分五个层次(物理层、链路层、传输层、驱动层和应用层)如图3所示。

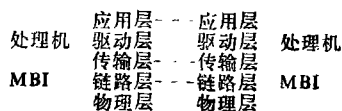


图3 通信层次

在这五层中,形成三个层次上的通信协议,如图中虚线所示。分布执行软件包括了传输层和

驱动层的通信功能,为用户任务提供了计算机间的通信手段。

分布执行软件驻留在每个结点机上并发运行。其功能是发送、接收消息,并实现消息同步和消息阻塞。消息的单位是数据块。

分布执行软件是不同机器中的任务之间相互通信和同步的主要手段。它实际上可以看作是一种共用信箱。每个数据块都与MBI中的一个子地址相对应。每个数据块都有一个逻辑名。一个机器可通过向数据块中写数据而达到与另一个机器中的任务通信的目的。反之,一个任务也可从一个数据块读数据,从而得到另一个结点上某个任务向它发送的数据,而数据块经过MBI在结点之间传送。

每个数据块都与一个发送子地址(源)及一个接收子地址(目的)对应。MBI子地址成为结点间通信的端口。根据1553 B 协议,一个结点最多只有 32 个发送及 32 个接收子地址。即每个结点最多只能收、发 32 对数据块。如果实际系统中的数据块数超过了 32 个,则可采用子地址复用技术,从而实现同一个子地址发送/接收多个数据块。系统为每个数据块设置了一个数据块控制表(DCB)。其中包含了许多对数据块操作的控制信息。数据块控制表分为读数据块控制表(RDCB)和写数据块控制表(WDCB)。

每个数据块控制表包含有该数据块的逻辑名、数据块所对应的子地址号、缓冲区指针及数据长度计数器等。在RDCB中还包含指向与读该数据块有关的任务指针。数据的发送分为两类。一类是周期性数据,另一类是非周期性数据。周期性数据的发送是通过时间驱动的,即每过一个解算周期,分布执行软件通知MBI把一批数据发向指定的结点。而非周期性数据是事件驱动的,即任务只要将数据放到相应的缓冲区,并将该数据块对应于矢量字中的相应位置位。MBI会自动地将数据发往指定的结点。

收信功能分为分信和收信。分信由MBI中断激活,完成消息的分解工作。主要是把数据复制到主缓冲区,若有任务正在等待该数据块,则将其唤醒。收信功能主要是把数据取到任务缓冲区,如果数据未到,则可根据情况选择是否挂起该任务。

四、系统软件与应用任务的接口

系统软件是为应用任务服务的。应用软件实际上是由分布在各台处理机中的若干应用任务构成的。这些应用任务在系统软件的支持下,相互配合,并发运行,完成预定的功能。系统调用命令是系统软件和应用软件的接口。应用程序通过系统调用命令获得系统软件的服务。用户在编写完应用程序之后,与系统软件一起提交给系统生成软件。生成软件把它们嵌入到系统软件之中,生成执行软件。然后使整个系统投入运行。每个系统调用命令都有返回值,应用任务中,可通过判断一个系统调用的返回值来确定系统调用的执行情况。本系统的系统软件是用C语言和8086汇编语言编写而成。C语言和汇编语言的优点使得本系统软件模块性较强,使用灵活,容易扩展,有较好的实时性。

本系统虽然是面向航空电子领域开发的,但它的应用并不局限于此。在一些大的、较为复杂的工业控制中,单道运行的程序已经不能满足要求,多任务并发控制、分布控制将成为未来控制的主流。该系统软件将在这些领域中有广泛的应用前景。(参考文献略)

(下转至 21 页)

- [2] Enhanced SCSI-Bus Interface Controller, Advanced Micro Devices, Inc. 1989.
- [3] PAL Device Data Book, Advanced Micro Device, Inc. 1989.
- [4] GAL Handbook, Lattice Semiconductor Corp. 1990.
- [5] Media Components, Vol. 1, Signetics Company, 1990.
- [6] Brooktree Product Databook, Brooktree Corporation, 1990.
- [7] Programmer's Reference Manual, Analogic Corporation, 1990.

A Multimedia Intelligent Interface System

Wang Xiyu and Zhang Xiang

Institute of Computing Technology, Academia Sinica, Beijing, 100080

Abstract: This Paper describes briefly a multimedia intelligent interface system, which is connected to a computer with standard SCSI bus for use as general video input and output device interface.

Keywords: System, interface, bus, multimedia, SCSI, intelligent.

(上接 25 页)

The Design and Implementation of a Distributed Real-Time Operating System

Gu Jianhua and Yang Zhiyi

Northwestern Polytechnical University, Xi'an, 710072

Abstract: This paper presents the design and implementation of an integrated-avionics-system-oriented distributed real-time operating system. It describes in detail the system structure, the task management, mail box management, interruption management, input/output management and distributed communication management in the system software.

Keywords: integrated avionics system, distributed real-time operating system, distributed communication.