

一种实用混合语言编程技术及其应用

黎洪生 (武汉工业大学自动化系)

摘要:本文介绍一种实用的高级语言和汇编语言混合编程方法,它能被有效应用于 PC 兼容机环境下的实时程序设计。

一、前言

在以 PC 兼容控制机组成的微机控制系统实时程序设计中,一般使用两种程序语言来编程:高级语言和汇编语言,它们各有特点。汇编语言直接面向机器硬件及过程通道,其高速和紧凑的实现程序,可以使得微机控制系统得到较好的实时性能,但是其对图形、数据处理、打印等功能的编程工作量及难度常使人望而生畏;高级语言以其编程调试方便,开发效率高,计算处理数据简单,所编制的程序人机界面精美而令编程者津津乐道,然而其对系统的实时性支持却不足。扬长避短将两种语言有机地结合起来用于实时程序设计的方法就是混合语言编程技术,已有大量文献对此作过研究介绍。所介绍的方法可分为二种,一是用数据文件作为接口中介,二是用行间调用语句。基本上都是基于对高级语言中调用汇编子程序的语法规则的解释和运用,并不能很好地解决实时程序设计中的问题。本文介绍一种实用的方法,其针对实时程序和编程语言的特点,基于一种实时程序的层次化模块结构,将实时程序中实时性强的控制模块和实时性要求不高的数据管理类模块分出,分别用汇编语言和高级语言编程,并在系统的支持下相对独立地并行运行,高级语言程序上层模块通过简单的接口控制汇编语言子程序模块并与其传送信息。这样有效地保证了系统的实时性能,并获得了用高级语言编程的好处。

二、实时程序与混合语言编程

1. 实时程序的特点

实时程序有别于一般的数据计算和管理程序,它的主要特点在于:

(1)实时多任务并发的处理。一个实时系统基本的工作是与外部 I/O 通道打交道,由于这些外部通道具有相

当的独立和连续运行性要求,因而要求实时程序应具备较好的多通道并行运行处理能力。

(2)程序模块执行的快速及时性。实时程序模块处理信息的来源和去向都为实时通道,这就要求与之相关的信息采集、算法运算处理、输出控制等程序模块具有较高的运行速度和较好的信息传送速率匹配性。为保证检测和控制品质,这种要求是严格的,常达 ms、us 级。

(3)控制模块执行的周期性。基于采样控制理论所构造的微机控制系统,对过程通道的处理一般是按控制周期定时进行的。这使得实时程序中的部分控制模块的运行方式具有周期性,即按一定的执行时间定时地启动、运行或停止。

(4)模块划分。实时程序模块可以根据其对实时性的要求不同较容易地划分为两类:实时模块(如控制通道处理模块)和管理模块(如参数输入、数据计算、显示和报表处理模块)。前者要求实时性高,适于用汇编语言编程;后者在数量上一般占实时程序的大部分,对支持语言的人机交互性能的要求较高,适于用高级语言编程。

2. 混合语言编程所要解决的主要问题

以 PC BASICA 和 8086 宏汇编语言为例,常用的调用方式如图 1 所示。在这里,汇编子程序模块被装入内存后,即处于睡眠状态,由高级语言程序中的调用语句唤醒其执行,其地位仅相当于一用户自定义函数而已,无助于达到上述的对实时模块的关键要求。

在实时程序中,运用高级和汇编两种语言编程所要解决的主要问题是:

(1)适当划分和安排程序模块功能,满足系统的实时性要求;

(2)装入汇编子程序模块的方式;

(3)高级语言程序对汇编子程序模块的控制调用;

(4)两种语言程序间的接口。

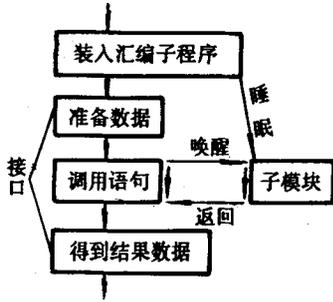


图 1 常用的调用方式示意

三、一种实用方法

1.实时程序模块层次化结构

在进行实时程序的模块功能设计时,将各模块根据

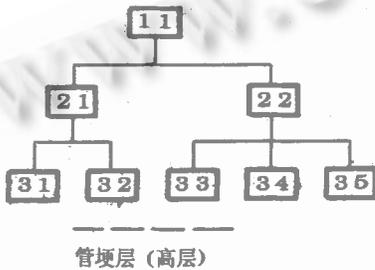


图 2 实时程序模块层次化结构

实时性要求的不同划分为两个层次上的模块:管理层和实时层,两层间的模块通过接口发生联系。

管理层: 管理层模块主要完成数据处理,人机交互等功能,实时性要求以显示、打印等能满足人的反应感觉即行,适于用高级语言编程实现。模块组织形式应用常用的菜单式结构。

实时层: 该层模块主要实现系统的实时及控制功能,即完成对时间要求严格或直接与外部过程通道打交道的任务。其编程语言以用汇编语言为好。模块结构形式应满足对系统的多通道并行及周期性运行要求。为此采用时间分片的分时技术来组织实时层模块。每一模块功能相对独立,一般仅涉及一个实时通道的全部或部分功能处理。

所构造的实时程序模块层次化结构如图 2 所示。

2.虚拟通道和实际通道

为便于管理层(高层)和实时层(低层)模块对过程通道的处理,提出实际通道和虚拟通道的概念。

实际通道: 物理过程通道,低层模块对系统实际通道的处理操作即为对控制通道和外设的控制实现,进行直接的 I/O 操作和硬件控制。可分为输入型、输出型和非输入输出型。

虚拟通道: 供高层模块所控制使用的(假想)设计通道。一个实际通道对应于一个虚拟通道。高层模块程序或语句所面对的仅是虚拟通道和常用的人机交互设备(如 CRT、键盘和打印机)。虚拟通道通过接口映射到实际通道。高层模块对过程通道的控制就象使用一条打印语句一样方便,其所作的就是对相应的虚拟通道发出命令而后从数据接口获得结果。因此在高层模块的高级语言编程时,将限制使用象 IN 或 OUT 这类语句,对控制

现场复杂的实时处理均由低层模块运行而通过对实际通道的处理来完成。

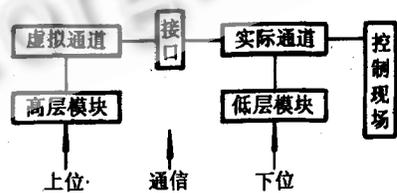


图 3 实际通道与虚拟通道

提出虚拟通道概念的目的,就是为了形成高层程序和低层程序在设计和运行时的相对独立。如图 3 所示,这类似于在一个系统中形成了控制(低层程序)和管理(高层程序)上下位两个子系统,由控制子系统完成系统的实时控制及其它实时功能,由管理子系统实现监控和数据处理等。而上下位之间采用公用 RAM 区通讯,并且不

存在对 RAM 区的竞争问题。

3.接口

对应于一个过程通道的低层模块与上层进行通讯的接口规约如下:

No.: flag; status; data

No.: 通道或模块号,一个字节。

flag: 执行标志,一个字节。该标志为本模块的运行控制标志。当其为 off(0)时,该模块停止运行;为 on($n \neq 0$)时,该模块将定时运行指定的次数 n , n 为 0FFH 时连续运行。该标志的操作权属于低层其它模块或高层模块。

status: 通道或模块状态标志,一个字节。该标志反映模块执行过程或完成后通道的状态,如成功、出错号等。其它调用模块通过该标志得到对应通道的状态和功能执行的情况。

data: 数据区,大小根据通道各异。用于与其它模块传输数据的缓冲区。

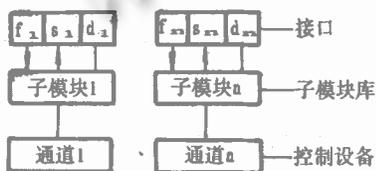


图 4 模块(层次)接口

如图 4 所示,实时层各(子)模块通过对相应实际通道的控制处理,将所获得的信息送入对应的状态和数据区,也从数据区得到高层(或其它)模块所传送的数据。所有模块的标志区和数据区之或集即组成了管理层与实时层程序(或模块)之间的接口。

为简单而便于应用,对接口规约另作如下规定:所有接口均设置在约定的存储区,该存储区不受系统(DOS 和高级语言支持程序)存储管理的约束,并且运行时明确接口区的物理地址。如可设置在单显系统的彩显缓存区、预留内存区等。

4.汇编子模块的编制与运行

编制:如图 4 所示,汇编子模块与外部的联系,一端为硬件设备,一端为接口存储区。根据其设计功能和接口规范即可编制出相应的子程序,图 5 给出了一个用于处理输出型通道的子程序流程图例。

组织:编制出所有汇编子程序后,按图 2 所示结构组

织各子程序模块。这里给出一个在工控 PC 单任务 DOS 环境下的简单方法。

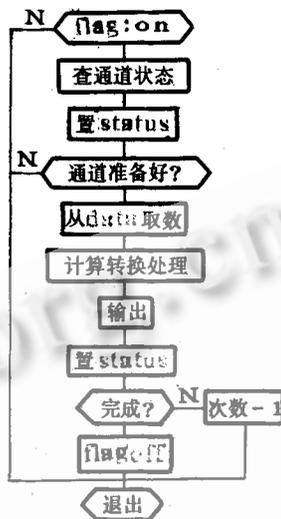


图 5 汇编子程序流程图例

在图 2 中,取系统定时通道中断时间为 T_0 ,如设计使 $T_i = n_i T_0, i = 1 \sim n, n_i$ 为正整数 $t_i < T_0, i = 1 \sim n$

则可以通过对单定时通道的中断次数的判断,转向执行各子模块 $S_i(i = 1 \sim n)$ 而方便地实现分时分片的任务。因为仅要严格保证部分子模块的执行周期,所以这种设置是较易实现的。

装入:在组织安排好各子模块后,按编写中断例程的方式组成一完整的程序,经汇编转换可得到一 COM 形式的实时层可执行程序文件。在系统启动或管理层系统初始化时,以内存驻留的方式装入该文件,确定其入口地址。

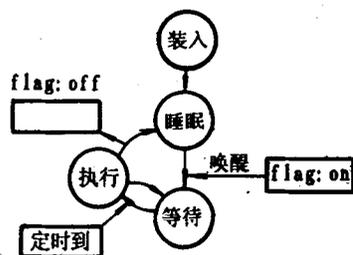


图 6 汇编子模块的状态转换

运行:修改 BIOS 定时中断向量,使其指向上述入口地址。这一步骤可在装入后随时进行。如在 PC / XT

主板机型中,可用 25H 号功能调用修改 1CH 号中断向量。此后,各汇编子模块程序均将以其设定的执行周期 T_i 分时作查询运行,如图 6 所示。

5. 高层程序对低层汇编子模块的控制

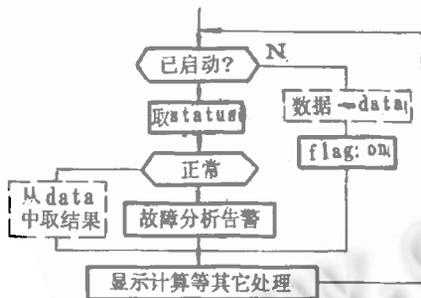


图 7 高层程序对低层子模块的控制调用例

在高层的高级语言模块程序通过改变接口中的标志 flag 即可实施对低层汇编子模块的控制,可通过 status, data 交换数据。如图 6、7 所示,如要从一个输入通道获取现场数据:高层程序在启动相应通道的低层子模块(置 flag 为 on)后即可进行其它(如计算、人机交互)处理;置 flag 为 on 的操作将使相应子模块从睡眠态变到等待态,等执行周期到,即从等待态变为执行态,分得一分片执行时间 T_0 ,完成与外设交换信息的功能;若该通道的

实时处理需要较长的时间,为与外部作良好的 I/O 配合,则处理将在该模块的几次执行周期完成;完成后子模块从执行态又变为睡眠态,并将结果送入对应的 status 和 data 区。而高层程序可通过在流程循环中查询 status, 并从 data 中取得结果数据。

由于在接口规范中规定了各通道模块的接口为约定的存储区,因而上述的高层程序的这种控制调用操作是简单的。如在 PC BASICA 中,用 peek 和 poke 语句即可实现。

四、结束语

我们曾在玻璃配料自动化等多个微机控制系统的实时程序设计中,使用这种混合语言编程技术,编程语言为 PC BASICA 解释版本和 8088 汇编语言。实用结果说明,与我们先前研制投运的同类型系统相比,用这种编程技术所实现的实时程序对系统有良好的实时性支持,与用全汇编语言编程的方式相当。而由于引入了高级语言编程,因而较容易地设计出了精美的 CRT 画面和彩色报表,大大改善了人机交互环境,并且增加了许多其它功能程序,如故障诊断、数据管理、调试帮助等,在很大程度上提高了实时软件开发效率。

参考文献:(略)