

大中型 MIS 的计算机软件结构模型设计

李爱国 苗云迁 法逸中 (江苏职工医科大学计算机中心 210029)

摘要:本文系统阐述了一个大中型 MIS 的计算机软件结构模型设计方案,从进程的功能划分、结构模式、运行控制以及数据结构设计等方面,论述了该设计方案中采用的主要关键技术。

一、系统进程的功能划分

进程是分时操作系统运行的实体,在考虑进程功能划分时,应使进程的功能分配合理,并最大限度减少进程间的交叉干扰以及资源等待,以提高整个系统的运行效率。在这种划分原则下,要求划分的每一进程在功能上要尽可能单一。系统划分为以下几大进程:

1. 总控进程

系统初始化,创建其他各子进程,管理和控制整个系统的运行;

2. 命令加工处理进程

管理所有终端用户的输入输出,对终端用户屏蔽所有操作系统命令,提高系统的安全性。改进程负责接收分析终端用户命令,并以信件的方式交有关进程处理;

3. 转储处理进程

定时转储整个系统的动态数据,所转储的动态数据供系统在故障恢复时使用;

4. 恢复处理进程

在系统故障后,依据转储进程储存的信息进行恢复处理;

5. 系统维护进程

监控整个系统的运行状态,为系统值班员提供各种必要的人工维护手段;

6. 网络传输进程

负责与其他网络节点的通信处理(实现信息共享);

7. 若干应用功能性进程

根据具体的作业流程合理地划分成若干个功能单一的应用进程,负责完成具体的应用事物处理。

二、系统进程的结构模式

在进程的结构上采用前后台为基本结构模式。进程被激活后首先进行一些初始化工作:建立邮箱(进程间通

信机制),并设置写邮箱注意 AST(功能模块的激活机制);然后处于事件等待状态(休眠状态)。前台(AST 例程)在收到信件后,对简单的事件处理直接完成,对需要较长时间的数据处理或文件 I/O 则排入后台处理队列,以事件标志触发后台处理。这样可缩短响应时间,提高运行效率。

进程的结构模式分为主程序和 AST 例程两部分。

(1)主程序:

```

BEGIN
  初始化工作
  建立邮箱
  设置写邮箱AST
  LOOP
    清事件标志
    等待事件标志
  REPEAT
    处理复杂事件
  UNTIL 复杂事件处理完
  ENDOLOOP
END

```

(2)AST 例程(中断处理模块):

```

BEGIN
  DO WHILE 信箱有信件
    读信箱信件并排入信件队列
  ENDDO
  LOOP
    IF 信件队列有信件 THEN
      挑出紧急信件作简单处理
      IF 有复杂处理 THEN
        复杂事件排队并置事件标志
      ENDIF
    ELSE
      设置写邮箱AST
    ENDIF
  ENDOLOOP
END

```

三、运行控制设计

1. 总控进程控制整个系统运行

总控进程控制其他各进程协调运行,其他各进程除受总控进程控制外,基本上是相对独立的并且是处于同一层次上。

在这种控制机制下,总控进程的可靠性最为关键,如果总控进程因故障退出,会使整个系统瘫痪。因此在功能设置上要使其功能比较单纯,把发生故障的可能性减到最小。总控进程在初始化整个系统后,主要负责各进程间来往信件的转投、进程退出时重建以及对整个系统运行的监控,总控进程不涉及文件 I/P 和具体数据处理,以防因外来数据诱发故障,所有的系统调用都留有出错

出口,以提高整个系统的安全性。

在提高整个系统的稳定性和有效性上,总控进程采取以下主要技术措施:

(1)总控进程转投各进程的来往信件,同时监视目的进程对这些信件的处理是否超时,若发现超时,向系统控制台显示警告信息,提醒系统值班员进行必要的干预;

(2)总控进程对系统公用资源(如公共数据区)的使用进行监控,超时告警,若发现某一进程长时间地占用某些资源,则强制释放这些资源,以免其他进程在等待这些资源时造成死锁而引起整个系统的瘫痪;

(3)对于影响系统性能的一些关键进程,总控进程采取实时升降其优先级的策略来辅助操作系统的进程调度。定时检查各子进程邮箱中等待处理信件的数目,发现信件积压时,实时提升该进程的优先级,疏通时再降到原来优先级;

(4)当某一子进程因故障退出时,总控进程总能立即再次创建该进程,并且能够重新转发哪些因该进程故障而未处理的信件;

2. 进程驱动源及进程间通信

每个进程在初始激活时,完成初启功能后都处于等待事件或休眠状态,再次处于激活状态的驱动源有:

(1)I/O动作完成,进程对输入/输出的处理,如命令加工处理进程接收终端用户命令、网络传输进程接收网络报文信息等;

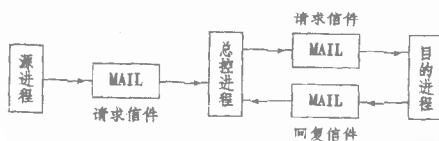


图1 信件来往示意图

(2)信件到达,信件是本系统各进程间通信的主要措施,分为请求信件和回复信件,请求信件是源进程请求目的进程完成某件事的信件,如加工数据、查询信息等请求,回复信件是目的进程在对请求信件处理后向总控进程返回的复信,是为了防止子进程在故障重建后遗漏对事件处理。信件来往如图1所示。

进程在对请求信件的处理上采用队列管理(急件队列和慢件队列)并且优先处理急件的策略,处理流程如图2所示。

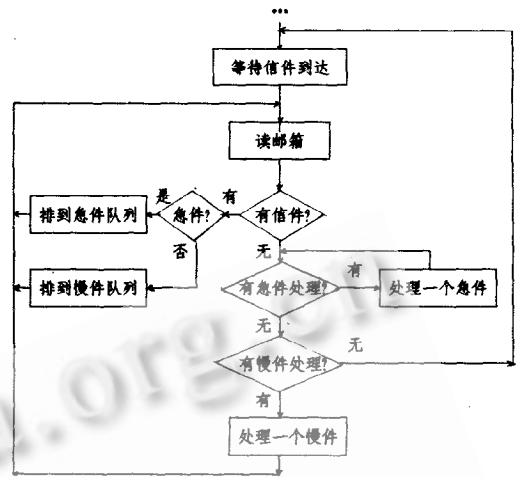


图2 信件处理流程图

四、数据结构设计

1. 数据结构形式

大中型 MIS 所要处理的数据量一般都非常庞大,在具体的数据结构设计中,根据数据用途的不同,主要采用盘文件和内存数据两种结构。对于那些必须保存留档的原始数据和打印输出数据,都以文件形式存盘;对于各种动态数据以及系统需要经常访问的固定数据,以共享数据的形式常驻内存,以提高数据的使用效率。

2. 共享数据的存取保护

为了确保共享数据的安全性和完整性,将数据与进程的可执行代码设计成相对独立的模式,这样不仅便利数据维护,而且所有进程的激活与退出都不影响数据的安全性。

为了进一步提高共享数据的安全性和完整性,采用资源封锁技术。将共享数据分为若干个相对独立的数据块作为数据资源的基本单元,并规定相应的资源名。进程在存取这些数据资源时,都必须申请资源封锁,根据情况的不同分为独占、保护写、保护读、共享写、共享读和空封锁,每一个数据块在某一时刻最多只能由一个进程修改,而其他进程仅能对它进行读操作。

为了避免进程间在数据资源占用上死锁,把资源按顺序编号,若一个进程同时使用一个以上资源,要严格按编号顺序逐个申请资源封锁。同时总控进程也有监控措施:强制释放那些被某一进程长时间占用的数据资源。

(来稿时间:1996年6月)