

TX 916.427

S 23

3

372780

S1240 程控数字电话交换系统

下 册

软 件 概 论

上海贝尔电话设备制造有限公司

人 民 邮 电 出 版 社

编 委 会 名 单

主任委员： 麦志强

副主任委员： 陈芳烈

委 员： 裘祖聿 冯大慈 朱其亮 顾重威

汤庭龙 李树岭 叶锦钊

前 言

S1240(本书正文中简称 1240)程控数字电话交换系统,自 1985 年引进到我国公众电话网以来,不到十年间,发展非常迅速。截至 1992 年年底,各省、自治区和直辖市已开通投产的新建和扩容局所已达七百多个,共三百余万线。销售总量达七百余万线。从而确定了 1240 系统在我国公众网中的重要地位。

1240 系统是国际上第一套采取全分布控制方式的大容量程控数字电话交换系统。其优越性已在世界范围内得到验证。中国电信网的运行实践经验证明,该系统的工作稳定可靠,无全系统瘫痪之虞,话务负荷能力及忙时呼叫处理能力较高,便于在线扩容。

ELC 型是 1240 系统的第二代产品,包括有大中容量局(M/L)、小型独立局(SSA)及远端用户单元(RSU)等系列化产品,可以满足各种类型的长、市话局需要。高达四万门的市话局和两万余路端的长话局已在国内公众电话网投入运行。容量更大的长、市话局也将陆续开通。

为了适应 1240 系统迅速发展的形势,满足国内用户培训技术人员和日常维护管理的需要,特组织编写本套书。过去我们也有过这类图书,但由于多系国外原文版或国内翻译的培训教材,其内容不尽适合国内人员的水平,因而不能充分满足维护管理工作的实际需要。

本书共分三册。上册“系统总体介绍”,着重讲述了 1240 的硬软件系统结构、性能、技术指标及工作原理等基本知识,使读者对系统整体有较完整和全面的了解;中册“硬件结构”,按构成电话系统的各种功能模块,逐个说明其结构、组成和工作原理,对关键部件进行了详细剖析,并提供了主要技术数据和元器件排列,以满足日常维护调测的实际需要;下册“软件概论”,着重对系统软件的基本知识作简要讲述,其内容包括 CHILL 语言,数据结构,数据库,操作系统及软件生成等。

由于 1240 系统软件和硬件结构比较复杂,且在不断改进更新过程中,而本书编写历时较长,在此期间,软件和硬件都有些变化。加上大部分写作人员缺乏编写书稿的经验,谬误或疏漏之处,恐所难免,恳切希望广大读者及交换技术专家们对本书提出改进意见,不胜感谢。

参加本书的编写人员,除特邀的南京邮电学院陈锡生教授、北京邮电学院的王钟馨副教授,宋茂强副教授外,都是上海贝尔电话设备制造有限公司的高、中级工程技术人员和培训部部分年青教师,计有:

叶锦钊	宋树成	邬晓勤	郭浩根
谢玉琴	陈秋华	刘鹏程	沈毓文
王力衡	赵晓春	章倩	李晶
倪国树	李建红	梁灿文	陈建华

参加本书审校、定稿的教授、主任高工和高工,都是邮电部门的资深专家,计有:

裘祖聿	俞维扬	杨明干	汤庭龙
朱其亮	王钟馨	叶锦钊	

在此谨向各位作者、审稿专家以及为本书出版作出努力的所有同志致以深切的谢意。

本书编委会

1993 年 4 月

目 录

第一章 1240 程控数字电话交换系统软件概述	(1)
1.1 模块化结构	(1)
1.2 消息	(1)
1.3 有限消息机	(2)
1.4 系统支援机	(7)
1.5 有限状态机	(10)
1.6 编程语言	(11)
第二章 1240 系统软件的组成	(12)
2.1 操作系统	(12)
2.2 数据库管理子系统	(13)
2.3 呼叫控制子系统	(14)
2.4 呼叫服务子系统	(14)
2.5 用户线和中继线硬件接口子系统	(15)
2.6 计费子系统	(15)
2.7 电话资源管理子系统	(15)
2.8 管理子系统	(16)
2.9 维护子系统	(16)
第三章 CHILL 语言	(17)
3.1 语言基础	(17)
3.2 语句	(29)
3.3 数据对象描述	(38)
3.4 过程与函数	(42)
3.5 模块	(52)
3.6 结构定义	(55)
3.7 指针	(60)
第四章 操作系统	(68)
4.1 操作系统概述	(68)
4.1.1 操作系统的组成	(68)
4.1.2 操作系统的环境	(69)
4.1.3 操作系统与应用程序的联系	(70)
4.2 进程管理	(71)
4.2.1 功能和组成	(71)
4.2.2 接口过程	(76)
4.2.3 作业调度	(83)
4.2.4 原语	(91)

4.3 消息处理	(95)
4.3.1 功能和组成	(95)
4.3.2 消息路由的确定	(97)
4.3.3 消息传送历程	(103)
4.3.4 过程的功能说明	(105)
4.4 时限服务	(110)
4.4.1 功能和组成	(110)
4.4.2 控制程序	(112)
4.4.3 绝对时限和相对时限处理	(114)
4.5 缓存管理	(122)
4.5.1 功能和组成	(122)
4.5.2 存储区的管理	(123)
4.6 其它模块的功能说明	(132)
4.6.1 差错处理	(132)
4.6.2 覆盖管理	(134)
4.6.3 与硬件有关的模块	(136)
4.6.4 引导装入与初始化	(140)
第五章 数据库及数据结构	(143)
5.1 概述	(143)
5.1.1 数据库的概念	(143)
5.1.2 1240 关系型数据库	(144)
5.2 1240 交换机的数据结构	(146)
5.2.1 数据关系	(146)
5.2.2 数据装载段(DLS)	(150)
5.2.3 数据操作指令	(161)
5.2.4 数据库管理系统	(167)
第六章 软件生成	(198)
6.1 概述	(198)
6.2 用户设计工程软件包的生成	(199)
6.2.1 环境库的生成	(199)
6.2.2 软件程序的生成	(199)
6.2.3 通用装载段和通用覆盖段的生成	(199)
6.2.4 人机通信软件包的生成	(201)
6.2.5 程序补丁装载段软件包的生成	(201)
6.2.6 不变数据库的生成	(201)
6.3 用户应用工程软件包的生成	(201)
6.3.1 用户应用工程硬件的设计	(201)
6.3.2 用户应用工程软件的设计	(202)
6.4 系统装载主磁带和系统装载数据磁带的生成	(202)
6.5 在线扩容的生成	(203)

第一章 1240 程控数字电话交换系统软件概述

1240 型程控交换机软件采用分布控制方式,它的软件都分布在控制交换机工作的所有微处理机中。软件系统由许多独立的模块组成,它们可根据交换机的不同要求来配置。在交换机扩展容量或增加性能时,软件的模块可以重新配置,程序发生故障时,软件模块也能自动地重新进行配置。

这种软件模块采用高级语言编写,且是全部分隔方式的独立分段程序,易于设计、编制、测试与维护,也较易修改以适应不同用户的新要求。

软件模块的设计,采用数据与程序完全分开的方式。对于不同要求的各种类型的电话局,其程序基本相同,而数据则每个局不同,只要改变局数据就能改变交换局的各种功能或信号方式(当然相应的硬件必须配合),这样就便于大量生产,易于查找故障,易于按用户的要求进行必要的修改。

系统还采用了虚拟机的概念。采用虚拟机软件结构技术,可使程序编制人员在处理虚拟机的高级程序时,不需要详细了解低级功能的执行情况。对交换系统来说更改硬件不会对系统产生影响。例如由于处理机硬件的发展而引起的变化,只需修改有关的操作系统而不影响应用程序的设计。

分散的模块化结构有许多好处,最主要的是:

- 系统软件不会因某一功能模块软件故障而全局瘫痪;
- 每个控制单元的软件功能较少,这样就减少了系统的复杂性,增加了可靠性;
- 系统能方便地扩容与调测;
- 忙时呼叫处理能力(BHCA)可以增大;
- 减少了单个处理机的负荷,从而使超负荷减少到最少。

1.1 模块化结构

1240 数字程控交换机软件由许多基本的软件模块组成,其最基本的模块就是称为有限消息机的 FMM。它是一个完成基本功能的实体。系统由几百个不同的 FMM 所组成,它们有各自的程序与数据,每个 FMM 都可以独立地进行编程和测试而不受其他影响。有限消息机 FMM 之间采用“消息”来传送信息。使各自独立的 FMM 能互相配合,完成系统的功能。下面就“消息”与“有限消息机”等概念逐个进行介绍。

1.2 消 息

消息(MSG)是具有一定格式的数据结构,FMM 之间只能通过消息来进行通信,诸如通

知 FMM 有一个事件发生(如号码接收)或要求 FMM 去执行一项任务(如连接话音通路)均需通过发送和接收消息来进行。对每一个消息的发送,系统将分配一个消息缓冲区(MSG—BUF),每一个消息缓冲区有 64 个字节,它由消息首部和消息体两部分组成。消息首部中将指出该消息的号码、优先级别、消息类型、消息体长度、消息发送者及消息接收者的有关信息;消息体内载有从一个 FMM 发送到另一个 FMM 的信息。

消息具有 0—7 八个优先级别。任一进程在运行过程中将输出有限的消息组合。对每一个消息的输出由操作系统作如下的处理:

- A. 分配一个空闲的消息缓冲区,并在消息首部和消息体内填入相应的内容;
- B. 将作业状态字(JSW)的相应位置位;
- C. 将该消息在相应级别“就绪队列”中进行排队;
- D. 当操作系统调度下一个作业(SNJ)程序运行时,按优先级别的高低,检查作业状态字,当轮到处理该消息时,才从相应优先级别的就绪队列中取出该消息送往目的地。

消息分为两类:基本消息和定向消息。基本消息即该消息发出的目的地未知,它要借助于操作系统的消息路由表来支援,才能被送到目的地(指接受该消息的进程)。而定向消息则是目的地已知的消息,它知道由哪个进程来接收,不必由操作系统来确定。当一个 FMM 要与另一个 FMM 的已分配用来处理某个任务的某个进程通信时,就使用定向消息。同一个定向消息可在不同的时刻送给不同的进程。

1.3 有限消息机

1.3.1 概述

任何系统所包含的全部功能可以分为两种主要的类型,即:异步或并行的工作、同步或顺序的工作,其原理和相互通信方法如图 1—1 所示。

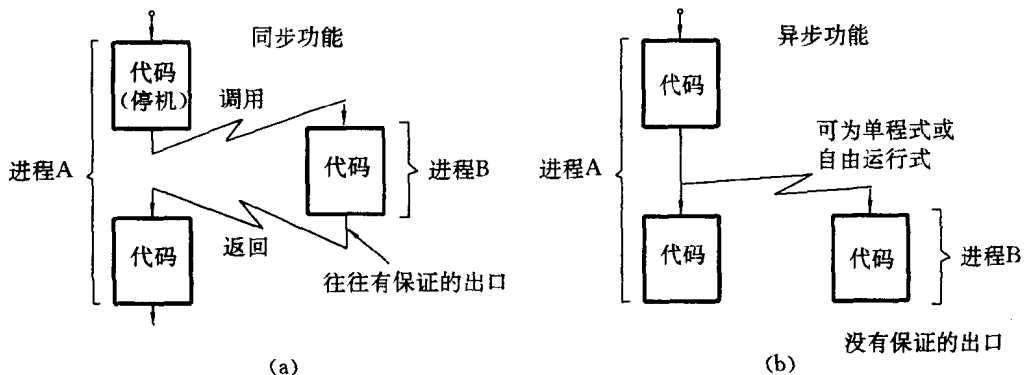


图 1—1 同步与异步功能

在图 1—1(a)中可看到一个同步软件进程 B 被启动后,在启动它的进程中必然会让进程 A 暂停,在进程 B 完成后又返回至进程 A。

而与同步进程不同的,如图 1—1(b)所示。在进程 A 调用进程 B 时进程 A 暂停,进程 A 仍在工作,也就是说启动进程与被启动进程能并行地工作,这种能并行处理的软件结构使启动进

程 A 和被启动的进程 B 可位于不同的处理机中。对一个给定的任务利用一台以上处理机并行工作,其结果是提高了系统的总处理能力。可以从图 1—1(a)中看出同步进程 A 与进程 B 置入不同处理机中不能提高总的处理能力。1240 程控交换机使用的多处理机就是基于这一原理提高系统处理能力的。

1.3.2 异步功能

在一个系统中异步功能的进程虽然必须与其他进程进行通信,但它们在系统中是独立的。该进程可以被系统中任何其他功能的进程所启动(有时也可被终止)。但启动(或终止)某一进程必须知道该异步功能进程的标志,异步功能的进程间进行通信就要通过消息,这种消息被引导到指定的进程。每个异步功能的进程都知道其进程所能接收与作用的消息组合。

异步功能的进程收到不属于其“已知组合”的消息时,就不予处理。

异步功能的进程也可向其他异步功能进程发送消息,发送的异步功能进程知道消息的标志及该发送消息经由的链路,但它并不知道连到链路对端的异步功能进程的标志及是否存在一个异步功能的进程。因此根据链路的连接方式同一消息可被不同的异步功能进程所接收,并起完全不同的作用,而发送的异步功能进程无需知晓。如在链路远端并未接异步功能进程,则该消息不予处理。

一个异步功能进程可以启动或终止其他异步功能进程,也可以启动(调用)同步功能进程,见图 1—2。

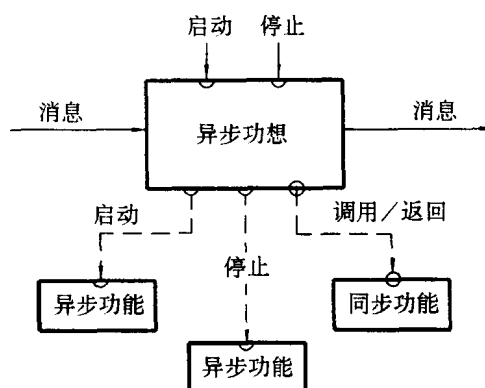


图 1—2 异步功能的相互作用

1.3.3 同步功能

同步功能进程是能被其他进程调用以完成一个特定任务的进程。在同步功能进程运行期间,调用程序处于暂停状态来等待该同步进程的结束。即由该进程来控制整个任务,待完成任务后再将控制归还并自行终止。

在异步功能进程中,存在着一个消息的时间差问题,参见图 1—3。

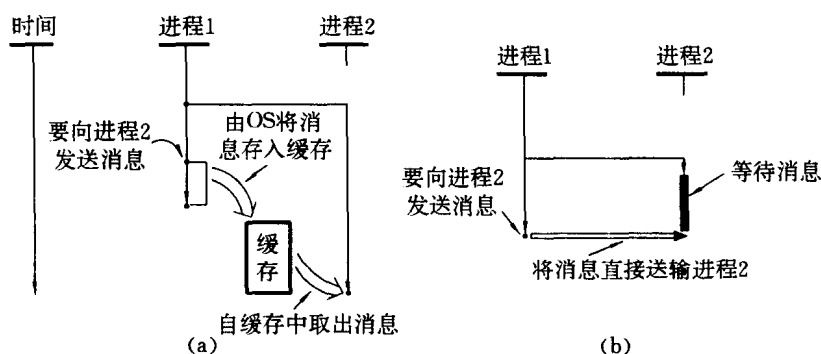


图 1—3 异步进程中消息的时间差

图 1—3 中进程 1 为调用程序,进程 2 为被调用程序。进程 1 要向进程 2 发送消息,有两种情况:

A. 如图 1—3(a)所示:发送消息时间早于接收消息时间,则由操作系统中的缓冲存储器管理程序将程序 1 发送的消息存在缓冲存储器中。待进程 2 需要时则询问缓存并取得消息;

B. 如图 1—3(b)所示:当进程 1 发送消息时,进程 2 已处于等待状态,就不需经过缓冲存储器,而直接发送。

从而可以看出,不论被调用多少次,也不管同时被几个进程所调用,同步进程必须是单程进程,每次调用时就建立一个功能环境(进程),返回时就破坏此环境。

与同步功能进程通信的正常方式是利用调用/返回方式,在调用时可将参数传送给同步功能进程,返回时又将参数传送给调用程序。

一个同步功能进程可不受限制地向其他进程发送消息,但一个同步进程等待或接收的只能是对其刚发出的消息和进行立即响应的消息(互控消息)。

在调用进程等待异步进程返回期间到达的消息,将被递延到返回后才受理。同步功能进程也可以和异步功能进程一样,启动和终止其他进程,也可以调用同步功能进程。要顺便提及的是整个电话交换系统就是一个异步功能。

1.3.4 有限消息机原理

有限消息机(FMM)是 1240 系统软件中一个最基本的软件模块,系统中有数百个 FMM。它们按不同模块,不同功能而分布在用户模块、中继模块、服务电路模块及其它各种模块中。

1.3.4.1 有限消息机的特点

A. FMM 是一个具有独立异步功能的实体能各自独立,并行工作,从外部看上去是一个黑盒子(即其结构不为系统其他部分所了解)。它必须被启动后才能起作用。启动时可以输入参数,可以接收有限的特定的消息组,也可以发出有限的特定的消息。它的功能完全由它接受与发送消息的顺序来定,它也可以启动其他 FMM;

B. 每个 FMM 具有有限的多种状态,而且可以在状态间转移。在某一状态下,必须依赖于外部输入特定消息而触发。才能使内部程序继续运行。在运行过程中可能会发出特定的消息,随后又可能引起状态的转移,进入新的状态。从而等待另一消息的到来,也可能在完成一定功能之后使程序终止;

C. FMM 之间只能通过消息来进行通信;

D. FMM 只有在被激活后才能运行;

E. FMM 工作状态及消息的发送与接收如图 1—4 所示。

图 1—4 可看到 FMM1 被启动后就进入初始化准备,再到初始化状态,等待消息。如有消息 X 被接收,就进行 X 运算,然后进入 S01 状态,又处于消息等待状态。这时若接收到消息 A,就执行 AA 运算并将发出消息 G 送到 FMM2,并使 FMM1 进入状态 S02。

如 FMM1 在 S01 状态接收到消息 B,将执行 BB 运算,并发出消息 H 到 FMM3, FMM1 又进入 S03 状态。

从图 1—4 可明显看出 FMM1 下一步做哪项工作完全由收到的消息及其顺序而决定的。

1.3.4.2 FMM 的结构

程控交换机中大量相同的部件并行而异步地工作着,如中继器、收发码电路、信号设备等。FMM 的逻辑能用来操作上述一个部件,还可用来并行操作一定数量的相同部件,其中每个部件需要不同的数据区。如每个多频互控接收器要有存放其地址的地方,对此,系统用一分用程序来定义操作的逻辑顺序,还要有一个与此程序有关的数据区来存放执行此程序所需的数据,

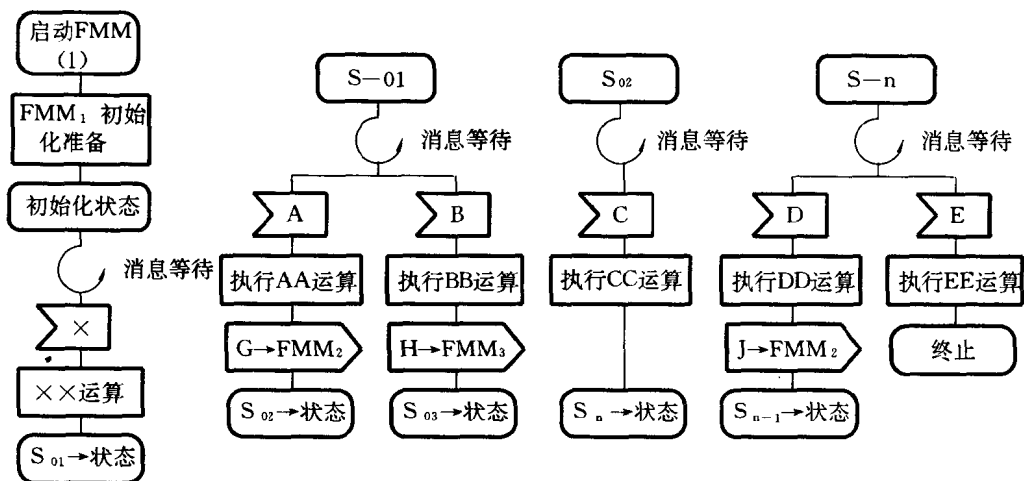


图 1-4

公用程序称作进程定义；使用其数据区来执行此公用程序称为一个进程。每个进程都有自己的数据区。

FMM 按逻辑分为两部分：即监视部分和应用部分，如图 1-5 所示。

监视部分是单进程的，其功能是控制应用进程。有如下特性：

- A. 它只包括单一的进程；
- B. 它只有一个在软件生产时预定义的入口点，操作系统核心经此入口点将基本消息送给此 FMM；
- C. 在进程执行等待消息时，此进程的再入口点就虚拟地存在；
- D. 若不是常驻内存的 FMM，可终止，常驻 FMM 不允许终止，对覆盖的 FMM，一旦终止就放弃其存储区。

监视部分有以下功能：

A. 接收送给本 FMM 的全部基本消息或吸收消化或将其转化为定向消息后，送给新建立的应用进程。如图 1-6 所示；

B. 需要时，在 FMM 内建立应用进程，并对应用部分的建立进程全面管理，包括必要的过负荷检测和响应，以及有关进程建立统计的记录和报告；

C. 对 FMM 实占资源进行资源管理。

FMM 应用部分有如下特性：

- A. 其进程必须在监视部分要求建立时才建立，所包含进程的静态或动态数量上限在工程设计时可以改变；
- B. 其进程能接收和发送定向消息；
- C. 其进程能发送但不能接收基本消息；
- D. 一个进程可以终止。

单进程 FMM 和多进程 FMM 消息接收和发送见图 1-7。

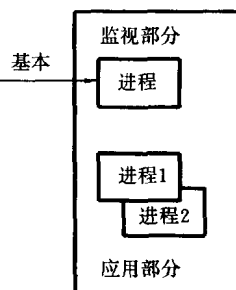


图 1-5 FMM 监视部分吸收消化基本消息

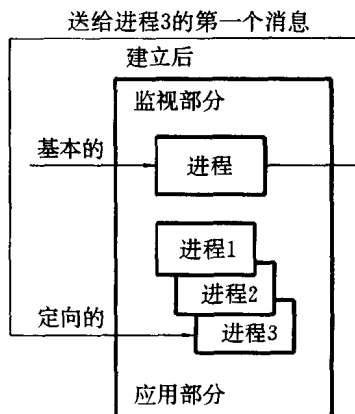
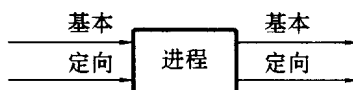


图 1-6 FMM 监视部分将消息转给应用部分

单进程FMM



多进程FMM

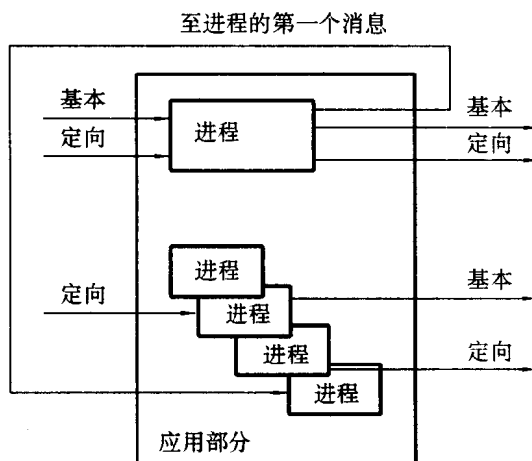


图 1—7 单进程 FMM 和多进程 FMM
消息的接收与发送

1.3.4.3 有限消息机的类型

FMM 按工作性质可分为单进程 FMM、多进程 FMM 和单进程多设备 FMM。

A. 单进程 FMM

单进程 FMM 由一个进程定义和一个进程数据组成,它既能接收基本消息和定向消息,也能发送这两种消息。

B. 多进程 FMM

多进程 FMM 由监控部分和应用部分组成,监控部分是一个单进程,它能接收和发送基本消息和定向消息,当它收到由其他 FMM 发来的基本消息后,就用定向消息送至应用部分的应用进程,它可根据需要用命令为应用部分分配相应的数据区,创建新的应用进程,当应用进程完成任务,监控部分就终止此应用进程,并归还数据区。而应用部分由一个进程定义和多个不同的进程数据组成多个应用进程。应用进程用来完成该 FMM 所规定的功能,当功能完成后,它即终止;应用进程可根据需要发送基本消息和定向消息,但只能接收定向消息,不能接收基本消息。

C. 单进程多设备 FMM

单进程多设备 FMM 只有监控部分,但它的数据区分为若干个块这些进程数据块数目固定,结构相同,且长期存在。一个进程定义可在不同的数据块上运行,进程结束后,进程数据块并不归还。

D. 进程状态及优先级

进程定义未配置进程数据时,进程尚未被创建,它处于空闲状态。一旦 FMM 接收了相应的消息,并为进程定义配置了进程数据,就创建了进程。进程创建后可具有四种状态。

a. 运行状态

进程被创建后,先进入运行状态,使进程在处理机上运行,执行相应的功能,包括向其它 FMM 发送消息,完成本 FMM 的状态转移。

b. 等待状态

每当进程完成了某一子功能需要等待消息时,进程就由运行状态转移到等待状态,只有等到所需要的消息来到后,进程才由等待状态返回到运行状态。

c. 中断状态

进程在运行过程中,若被中断,就进入中断状态。被中断的进程只有重新调度,才能恢复运行。

d. 封锁状态

如果所需要的存储器资源未能满足,则进程就暂时进入封锁状态;在获得资源后,再恢复运行。

当进程被终止时,就返回空闲状态。

1.4 系统支援机

1.4.1 系统支援机概况

上面介绍了 FMM 是通过发送一个消息来调动一个进程的,这种方式有利于系统的模块化。但是通过消息来调度进程速度是很慢的,传送每一个消息都需要一定的处理时间。

为了解决实时性很强的软件,引入了系统支援机(SSM)的概念。要求它能对特定事件立即作出反应,因为它是与硬件直接有关的软件,为了使其响应块,采用汇编语言编写。SSM 主要用于频繁的重复性的工作如扫描等。这样,使处理机的开销大大减小。为了将应用软件与硬件完全隔开,对 SSM 的调用都由操作系统统一管理。SSM 只有在系统的实时性需要时才用。一个 SSM 不能调用另一个 SSM 的数据。

1.4.2 系统支援机的过程

一个 SSM 可包括一个或者几个过程(PROCEDURE),它们可为一个 FMM 或一群 FMM 提供支援功能。SSM 可有以下几种过程,即接口过程、中断过程、时钟中断过程及事件处理程序。下面将逐一介绍。系统支援机(SSM)的基本结构如图 1—8 所示。

1.4.2.1 接口过程

它是完成一定硬件功能的子程序,是 FMM 的延伸与扩充,就如 FMM 的一部分。它可以发出消息,但它所接收的消息只能是它发出消息的回答。它是唯一能被 FMM 通过操作系统调用的 SSM。一旦被 FMM 调用,在接口过程运行完毕,就能返回调用的 FMM。

1.4.2.2 中断过程

它是处理外部设备中断如磁盘等的中断服务程序,当外设产生一个中断时,由操作系统启动调度执行相应的中断驱动过程,并负责关中断处理外围部件。完成任务后再开中断,并等待下一部件中断来调用。

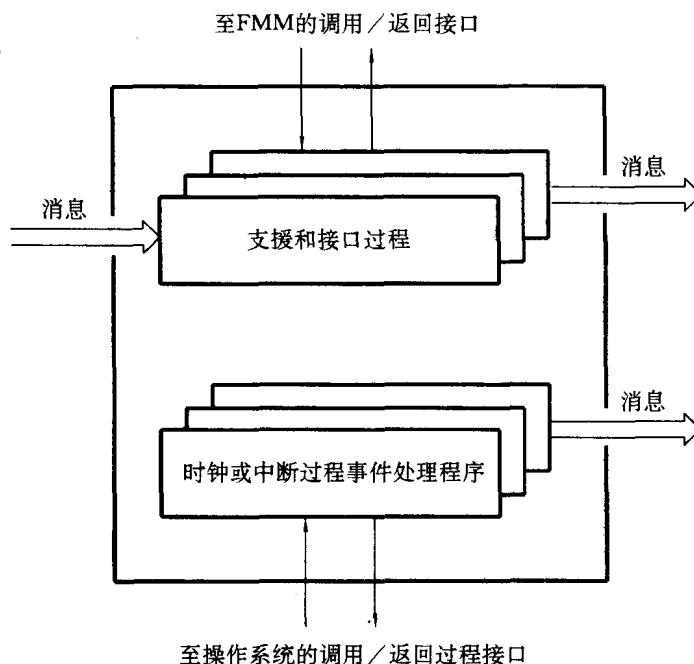


图 1—8 系统支援机 (SSM) 的基本结构

1.4.2.3 时钟中断过程

它是周期性响应“时钟中断”的中断服务程序,进行对硬件设备的扫描或查询,例如对用户线和中继线的状态进行监视扫描。周期性的时间从 5 毫秒至 300 毫秒,一个时钟过程一旦被调度执行,就一直执行,直到完成了任务,然后等待再一次被调度。

时钟过程在关中断情况下运行,即在运行过程中不允许再被中断,时钟中断运行时间小于或等于 1 毫秒,它不能接收消息,只能用来发现事件,登记事件,作一些必要的简单处理,对来不及处理的事件,先在属于操作系统的事件标志表中对相应的单元置“事件标志”,然后由事件处理程序作进一步处理。

1.4.2.4 事件处理程序

时钟中断过程与中断过程均是在关中断情况下运行的,必须尽量少占处理机的时间。这些过程将它们来不及做的工作延给事件处理程序来做。推延的方法是在操作系统的事件标志表中相应单元置“事件标志”。该标志有一个相应的事件处理程序,它是在开中断情况下运行。允许被中断,运行时间小于 10 毫秒。它只能发出消息,不能接收消息。事件处理程序一旦被调用必须连续运行,直至结束。

1.4.3 系统支援机过程的特性

1.4.3.1 SSM 接口过程特性

- A. 它是包含在一个 SSM 里的一个过程;
- B. 同一个 SSM 里可存在不同的接口过程;
- C. 它可作为一个 FMM 到 SSM 的接口。如一个 FMM 可通过接口激活硬件命令或要求该 SSM 所知悉的信息(如几点几分);
- D. 接口过程可从概念上被视作包含在 FMM 内,能被该 FMM 调用的一个过程,虽然在实际应用中 FMM 激活接口过程是不一样的;

E. 它能以 FMM 的名义发送消息；
F. 它能以 FMM 名义等待和接收互控消息传送的消息。如它发送一个消息并等待立即回答；

G. 可用监督程序与同一 SSM 的其他部分进行通信；

H. 可以使用 OSN 性能。

1.4.3.2 SSM 监督程序的特性

A. 容许安全进入公用 SSN 数据；

B. 在关中断下运行；

C. 为该 SSM 的其他部分所激活。

1.4.3.3 SSM 中断过程的特性

A. 当识别其相应部件来中断时,即开始执行；

B. 在关中断下执行；

C. 可发送消息；

D. 不能等待；

E. 不能接收消息；

F. 可用监督程序与同一 SSM 的其它部分进行通信；

G. 不能利用时间服务(如超时)来对部件进行安全监视。此功能必须由相关 FMM 来做；

H. 与同一 SSM 的其它部分和控制 FMM 异步运行；

I. 可在事件处理程序中产生一个事件；

J. 执行时间短少于 1 毫秒。

1.4.3.4 SSM 时钟中断过程的特性

A. 它被周期性投入执行,执行周期的范围是 5 至 300 毫秒；

B. 在关中断下执行；

C. 能发送消息；

D. 不能等待；

E. 不能接收消息；

F. 可用监督程序与同一 SSM 的其它部分进行通信；

G. 它也不能利用时间服务(如超时)来作为部件的安全监视。此功能必须由相关 FMM 来做；

H. 可对事件处理程序产生一个事件；

I. 执行时间短,少于 1 毫秒。

1.4.3.5 事件处理程序的特性

A. 发生了对本 SSM 的事件后即投入执行；

B. 在开中断下运行；

C. 可为时钟或部件中断所中断；

D. 可发送消息；

E. 不能等待；

F. 不能接收消息；

G. 与同一 SSM 的其它部分以及相关的 FMM 异步运行；

H. 执行时间可达 10 毫秒；

I. 在进程前执行(即高优先级)。

SSM 和 FMM 相互间配合工作如图 1—9 所示。

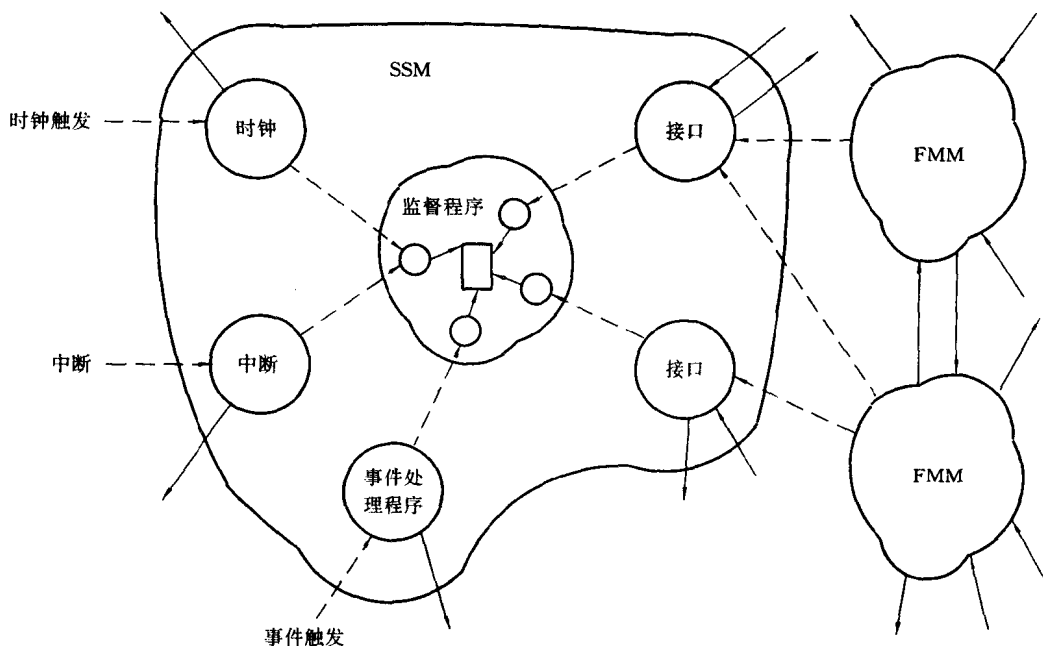


图 1—9 SSM 和 FMM 配合工作图

1.5 有限状态机

1.5.1 有限状态机概念

程控交换机的很多程序由顺序的动作来组成,由一个状态进展到另一个状态,这些状态往往用以表示一个实时中断或程序的一个主要状态。有限消息机处于某种状态一直等待到产生一个输入事件,例如接受到某一个消息,消息的接收就启动一个子程序来执行某些任务,然后根据这些任务的结果,使有限消息机(FMM)进入一种新状态,这些状态的组合可以称为有限状态机(FSM;FINITE STATE MACHINE)。

1.5.2 有限状态机状态变化示例:

图 1—10 作为典型的呼叫处理进程的状态变化示例来说明 FSM 的特点。设该进程建立后处于 0 状态,等待消息 A、B、C 的到来。假设接收到消息 C 则执行 C 运算,执行的结果发送消息 P,并处于状态 3,并再次等待着消息 Q,一旦收到 Q 后就执行 Q 运算,然后转为状态 4,并等待消息 D 的到来……。

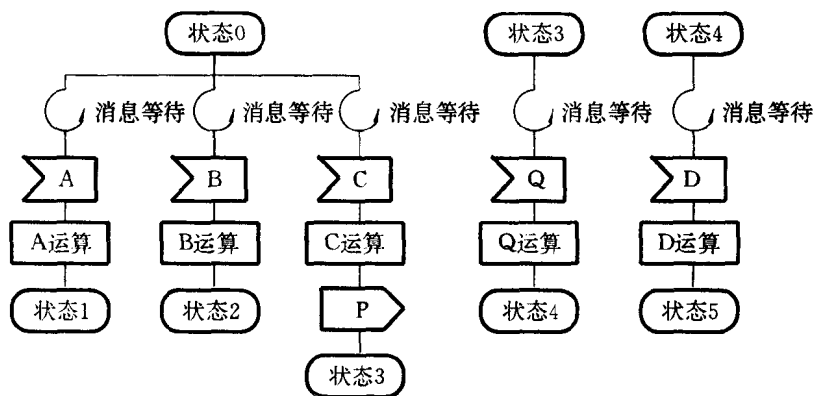


图 1—10

1.6 编程语言

S1240 系统的基本编程语言是用国际电报电话咨询委员会公布的程控交换机高级语言 CHILL。为了简化程序在 FMM 中还使用了面向问题的 (POL) 语言。POL 语言在源程序中, 可与 CHILL 并列使用, 然后可由编译系统把源程序展开为纯 CHILL 语言, 再由 CHILL 编译系统展开为 8086 汇编语言及机器码。

对于有时限要求的与外部设备配合的程序, 如好些 SSM 则直接用 8086 汇编语言编写。

(本章由叶锦钿编写, 朱其亮审)

第二章 1240 系统软件的组成

1240 系统软件就可以分成 9 个独立的功能子系统。

- 操作系统
- 数据库管理子系统
- 呼叫控制子系统
- 呼叫服务子系统
- 用户线和中继线硬件接口子系统
- 计费子系统
- 电话资源管理子系统
- 管理子系统
- 维护子系统

每个子系统都能进一步分成几个有明确定义的功能块,除了操作系统以外,这些功能块在应用软件中由 FMM 或 SSM 来实现。

2.1 操作系统

操作系统(OS)为不属于应用软件的软件子系统。

操作系统的主要功能就是装载程序,控制初始化和支援应用程序。操作系统分成 4 个部分:

- 输入/输出控制系统支援(IOCS)
- 支援基本功能
- 对虚拟处理器进行初始化和恢复
- 提供公共服务

2.1.1 支援基本功能

本程序执行如下软件的基本功能:

——操作系统核心(OSN)用来调度程序运行的进程,处理各类中断,管理各进程,执行定时检测和告警,分析故障,管理存储器以及建立信息路由。

——操作系统核心包括:

- * 控制和处理各种控制单元中软件错误及硬件故障;
- * 管理及控制所有与时间相关的功能的时限服务;
- * 执行对存储器资源的调度和缓冲区管理;
- * 控制 SSM 调度和执行以及 FMM 的进程管理;
- * 支援 FMM 之间通信的消息处理,确定消息的目的地和传送消息的通路,并按优先级