

# 实时操作系统(RTOS)——用于宇宙航行 实时控制的扩展操作系统(OS/360)

譯文資料

(1)

长 沙 工 学 院 资 料 室

一九七一年十二月

# 实时操作系统(RTOS)——实时宇宙航行控制 的扩展操作系统/360(OS/360)

J.L. 約翰斯頓 著

国际商业机器公司  
得克萨斯州·休斯敦

## 引言：

实时操作系统/360(RTOS/360)，即标准国际商业机器系统/360(IBM系统/360)操作系统(OS/360)，的修改方案，是为了配合NASA(美国航空与宇宙航行局)的阿波罗宇宙飞行中的实时计算机综合组(RTCG)而由IBM的联合系统分队(FSD)发展的。实时操作系统/360是一种实时的、多任务、多工作的操作系统，它扩充了操作系统/360的基本特征，增加了如下的一些附加特征：

- 处理实时数据
- 使程序编制员使用简单
- 保证快速响应系统的活力(要求范围从十分之一秒到一秒)
- 增进效率
- 管理专外设备，OS/360中没有这些设备
- 提供纠错系统
- 增加任务库的新内容

操作系统/360中存在这些特征并不妨碍它的基本能力，就是说现在IBM发表的操作系統/360不論是以标准的或非实时的执行方式来操作，其所有的设备对于实时操作系统/360都是通用的。

国际商业机器公司的联合系统分队休斯敦操作室所研究的并且加到构造RTOS/360的OS/360中的几种主要功能范围如下:

- 独立的任务管理
- 系统工作能力
- 排队的管理
- 数据和时间的例行处理
- 时间管理
- 实时输入/输出控制系统
- 数据列表
- 显示形成的语言(DFL)
- 实时连续操作
- 大磁心存贮分配
- 记录
- 模拟输入控制
- 加快时间
- 纠错程序
- 事后处理
- 休斯敦自动传输线段的奇偶检查
- 统计推断系统
- 任务计数系统
- 多种工作

#### 实时操作系统的环境

虽然实时操作系统/360能够用于多种用途和多种计算机系统组式,但仍需要优先讨论它的功能范围,这是因为我们要建立这样一个环境,即实时计算机综合组(RTCG)•实时计算机综合组硬设备和实

对计算机综合组使用程序在此环境中进行操作。

### 实时计算机综合组：

实时计算机综合组是地面基地的计算和数据处理的综合体，用于美国宇宙局的载人宇宙飞程序。它包括计算机设备，组合的外围设备和程序部分，以便监督和保持——实时地——阿波罗的发射、模拟和训练练习。(I)

实时计算机综合组是美国宇航局得克萨斯州休斯敦城发射控制中心(MOC)的核心部分。MOC的飞行控制人员要监督载人宇宙飞行的每一个阶段，如发射进入轨道、重返和着陆。在去月球的飞行中，飞行控制人员也要监督和支持宇宙航行人，在他们飞往月球、降落于月球表面、起飞及与母船会合和返回地球过程中都要这样。

飞行控制人员所需要的用以监督飞行和对派遣问题做出决定的那些信息由实时计算机综合组供给。这简单的意思就是飞行控制人员坐在休斯敦的控制台前，实时地获得正确的信息，诸如每一个船上系统的情况、宇宙航行员的条件、在任一要求的时间内，在40小时之内，宇宙航行人空间的位置、或者宇宙飞船上或宇宙航行人任一计划内的演习动作的效果等。

在发射飞行期间，实时计算机综合组要做很多事情。比较重要的或比较共同的一些要求包括以下各点：

- 发射的时候处理雷达数据，并给飞行控制人员提供当时的位置和速度。
- 给飞行控制人员提供宇宙飞船到达还是没有进入轨道的信息。
- 处理遥测技术数据，并给飞行控制人员提供关于生命的信息，宇宙航行人环境控制系统中氧气的剩余量。
- 根据雷达数据计算宇宙飞船的轨道。

- 予告宇宙飞船在著来某些时刻的位置。
  - 计算宇宙飞船如何和何时必须完成一种特殊的动作去改变它的轨道特性。
  - 给宇宙飞船上的阿波罗制导计算机计算 惯性导航信息。
  - 处理雷达范围的数据，告知飞行控制人员宇宙飞船是在正确的月球转换飞行轨道上，如果不是，需要采取什么动作使它回到正确轨道上。
  - 重返期间监督阿波罗制导计算机并予示宇宙飞船着陆点。
- 除了上述这些任务和在一次典型阿波罗派遣飞行期间要完成数千以上的任务外，实时计算机组在飞行控制人员和全体有关人员的训练中也有关键的任务。

为了完成实时计算机综合组的各项不同要求，有五台 IBM 系统 / 360 七十五型计算机，其中每一台都分配不同的任务，而且这些任务在任意时刻都能进行调换，这是在设计实时计算机综合组时就注意了的。这一组合成一体计算机能允许美国宇航局在同一时间进行两个真实的派遣飞行，在同一时间进行两个模拟派遣飞行，或者在同一时间进行一个模拟飞行和一个真实飞行。图 1 是实时计算机综合组，它说明下述一种组合形式进行工作的五台系统。在给出的这种组合中，网络数据从发射控制中心的通信指挥及遙控 (COATS) 之一进入实时计算机综合组，然后，数据被送到实时计算机综合组的飞行操纵计算机 (MOO) 里，它处理所有的派遣实时处理任务，而动态备份计算机 (DSO) 则完成多余的处理任务，并准备去起如同 MOO 的作用，假如需要的話。模拟和训练演习中，阿波罗教练员不论是在 MOO (飞行控制中心) 或者在肯尼迪角，都是在相同的“飞行操作控制室” (MOOR) 之一的封闭室内。(其它的 MOOR 用于进行中的飞行)。一个做模拟用的计算机包含一个使用程序，该程序产生模拟网络数据；其它计算机则用作模拟

操作计算机。第五台计算机是两种练习的备用计算机，然而，并不是没有用的，它能进行任务库的修改，以适应未来使用程序的发展。

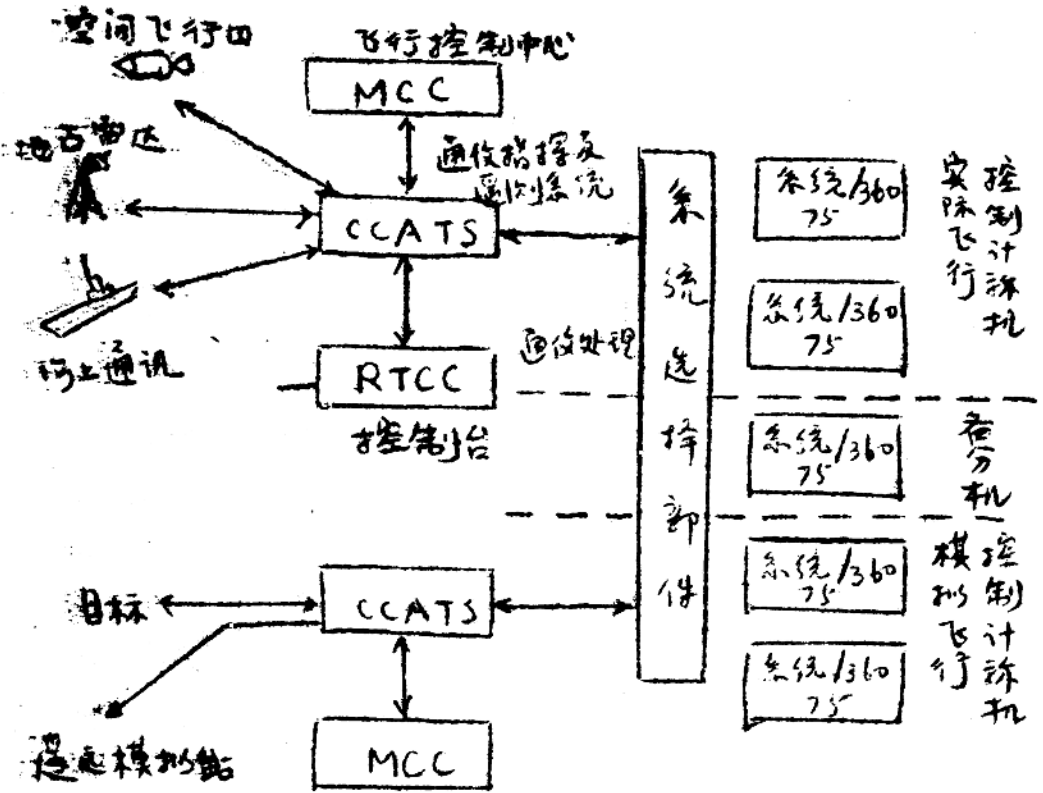


图 1-- RTCC (实时计算机组)



## 硬设备构造

系统/360硬设备构造有若干种，用于发展和实施国家宇航局关于实时使用方面的计算机系统任务。每一种构造均由单台的RTOB/360系统（实时操作系统）来对付。RTOC五台计算机系统每一台的构造本身包括一台有一百万二进制组主存贮器（IBM2705）的360/75计算机（见图2，维持飞行的系统/360型75）。

IBM2361大容量存贮器（LOS）用作扩大为四百万个二进制组的存贮器，也用作从IBM2314磁盘中调数据和程序的缓冲装置。IBM2701给MOOR和RTOC内部的数字显示（D/TV）系统提供所需要的快速响应接口。大量的数据和控制信息的实时采取和发送，用IBM2902多路转接器（MLA）来完成。一台卡片输入/穿孔输出机，一台IBM1443打印机，三台IBM1403打印机，两台IBM1052控制台和八台磁带机构成一个完全的组合。

其它的系统360/75组合，主要是用于模拟操练。除了前一种组合所提到的这些设备外，这种75型组合提供特殊的阿波罗模拟处理机通道（ASPO），它接收下述设备来的数据，即复合通道和分配器（MDD），IBM2260显示器和IBM2844（它用作IBM2314磁盘的控制器）。有几种不同的系统360/50组合也是用RTOC的RTOB/360来配合的。



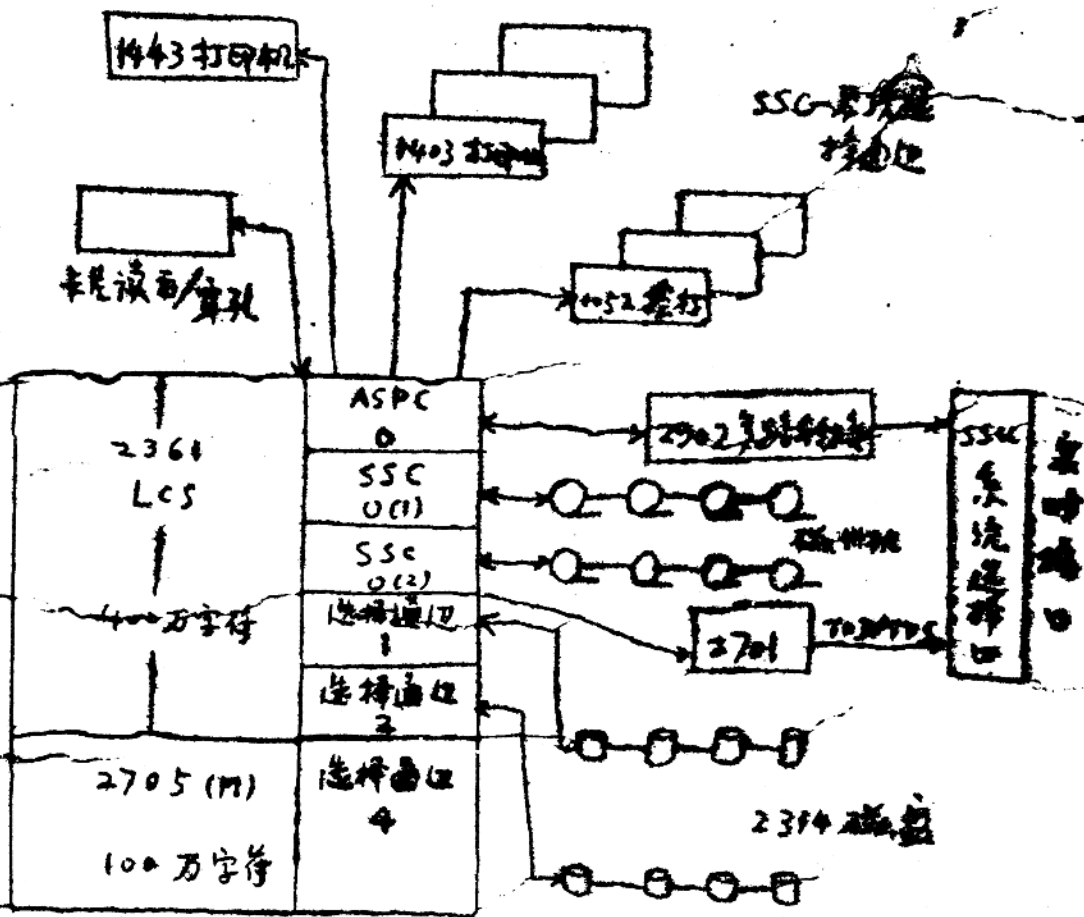


图 2 维持飞行的系统 360/75

应用

这些不同的组合中所执行的程序部分包括

- 1 阿波罗飞行任务系统。

- 2 地面模拟机器系统
- 3 动态网络数据产生系统
- 4 模拟校验和训练系统
- 5 操作准备和可靠性测试系统。

我们现在已经说明了RTOS/360的环境，即RTOS，RTOS硬件组合和RTOS在RTOS/360控制下进行实时处理的应用。记住这个环境，我们现在就能转而叙述将OS/360扩展为RTOS/360时的各种不同的功能和特征。首先，让我们看一看功能。

### RTOS/360的功能。

#### 独立任务的管理

OS/360中，所有处理任务的完成都是与规定的任务的“工作单元”有关。所谓任务，既不是磁心存储器中的程序，也不是有关任务的程序段。一项任务是一个工作单元（程序和数提）要素一定的东西（CPU等）以完成它的功能。只有当“任务控制块”（TCB）已经构成而且操作的管理部分已得知它的位置的时候，一项任务才能存在。“任务控制块”（TCB）包含有诸如数据指示器等的信息（I/O），执行任务有关的程序段，以及任务的优先级等等。OS/360中，“多道程序”这个词被“多重任

务<sup>■</sup>所代替，然而其意思是一样的，就是说很多任务是异步进行处理的——按照系统的要求，各种逻辑路线是借助於CPU进行开关的。

(图3，任务，一项任务的图解)

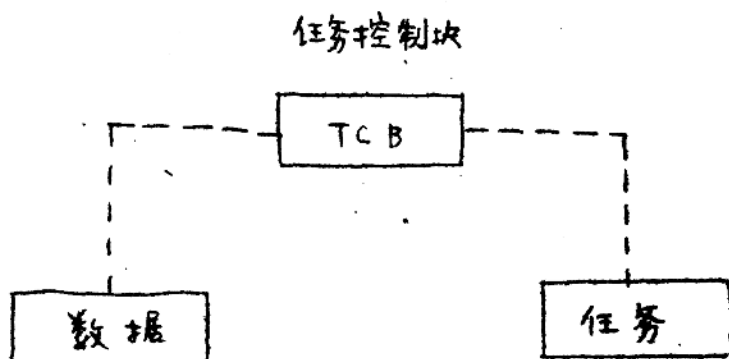


图3——任务

OS/360任务的一些特性在其功能方面和实时系统所要求完成的工作类型没有多大关系。OS/360任务为了存在而要求其母任务的存在，也就是说它依赖于它的产生者。OS/360这个概念在RTOS/360中已被引伸或包括不依赖于它的母任务而独立的那些任务。这样就出现了非独立和独立任务之间的区别。(非独立任务与OS/360任务相同)。所以，独立任务就不为其存在而要求其母任务的存在。

一项独立任务的特性怎样才能使自己更好地适合于实时系统呢？

第一，实时系统必须能够迅速有效地接收和处理各种各样的数据。RTOS/360中，对于每一种类型的数据都可规定一项独立任务，这里对全部时间内接收数据工作，甚至是在传输数据的速度较低或随机的情况下，也都适用。非独立任务和任务之间的主要区别是当系统中已有数据进行处理的时候，独立任务将继续存在。在这段时间内，它是“暂停的”。OS/360任务为了存在，至少需要一个输入模型在工作。每一独立任务，在称为“源表”的主磁心中分配给一个区域。这是一种专用区，能被此任务的程序所使用。通常，信息是存储在这样的区域里，它是从处理前一个数据中得来的。以此方法，任务能够“记住”整个暂停时期的信息。当系统为一项独立任务而接受数据的时候，这些数据是以“请求”的形式送给任务的。每一请求有它自己的优先级，优先级反过来就说明了处理这个请求的急迫程度。对于OS/360任务说来，只是它的某任务有优先级。如果独立任务正在处理一个请求而另外一个请求已经产生的话，则新的请求就按其优先程度进行排队等候。当任务把优先程度较高的请求或早一点的请求处理完后，排队等候的请求再被送入。

当独立任务在执行的时候，应给以唯一的保护键。对于由独立任务在处理请求时所产生的全部非独立任务都应给以这种保护

键。所以，对某一项独立任务或其下行任务中的程序工作时，将给予保护，免受其它在执行的独立任务或其下行任务所控制的所有程序的干扰。因此，非独立任务均配给与母任务相同的保护键，OS/360中一项工作步骤的所有任务都有这样的保护键。这个在大型实用多道程序系统中并不实际，其中有很多任务，处理各种各样的数据。独立任务能保证这种唯一的保护键将被指定为唯一的功能。

图4，OS/360任务的结构，表示OS/360中作为工作步骤而操作的任务的逻辑结构。这种结构是明显的金字塔形结构。所有任务都或者直接地或者间接地依赖于“工作步骤任务”。“工作步骤任务”能产生非独立任务（子任务），非独立任务依次又能产生依赖于它的任务，等等。所有任务都争夺系统资源（CPU，I/O，等等），同时OS/360把这些资源按其优先程度分配给每一任务。

图5，RTOS/360任务的结构，表示RTOS/360中作为工作步骤而操作的任务的逻辑结构。可以看出，这里增加了新元素。“工作步骤任务”及其子任务和OS/360中的一样，而每一独立任务则成为另外一组任务的基础，这组任务独立进行操作，属于和平行于“工作步骤任务”。这种结构比得上OS/360中的多工作结构，该结构每一独立任务相似于每一主动工作的“工作步骤任务”。然而，RTOS/360中所有的独立任务及其子任务是以单一的

工作步骤起作用的，而且那个工作步骤里的所有任务是按照优先程度来接受系统的资源的。

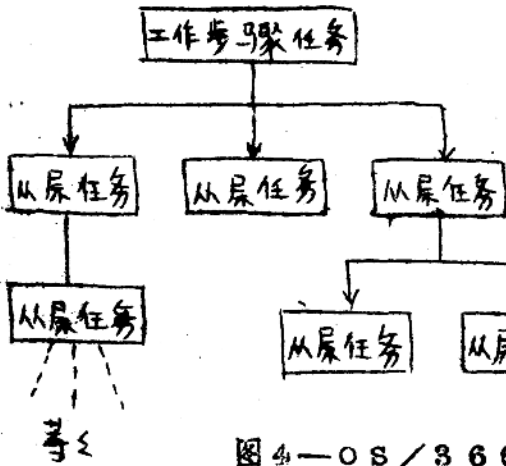


图4—OS/360

任务的结构

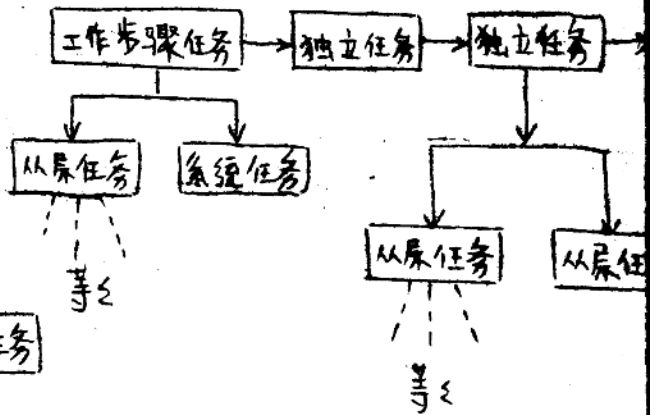


图5—RTOS/360

任务的结构

### 系统任务的能力

处理实时数据的过程中，会发现有许多工作任务与现在的任务没有关系，其完成时间与现存任务是非同步的。这些任务确实是系统的任务。因此，RTOS/360中处理系统任务的能力有所发展。系统任务是为RTOS/360的管理人员或“用户提出的任务”，如输出信息和记录实时输入数据。系统任务可在1/25的系统剩余时间内产生和返回，这个剩余时间不论是OS/360规定的非独立任务或者是RTOS规定的独立任务都有所需要。实时环境中履行所需要的系统服务功能，而减少剩余时间在高速度、实时数据处理的CPU

临界时期中极为重要。剩余时间中的差别是由于下列原因造成的：

- 系统任务要求的所有控制块，为了有效地使用，已被预先分配和预先启动。

- 每一系统任务的入口是一个绝对地址，而不是象非独立任务和独立任务的情况那样，输入模型名称。

### 排队管理

如果独立任务正在处理一项“工作请求”，则该任务的所有其它“请求”都必须由系统掌握住，直到该任务已准备好开始处理新请求时为止。因此，RTOS/360，必须构造并保持一个工作请求的“排队”。该排队的等候由执行的独立任务进行处理。有关每一请求的信息，保持在“实时排队单元”（RTQEL）中。（图6，独立任务及RTQEL，表示一项独立任务及其RTQEL的逻辑结构，该RTQEL正等候处理）。每一个在执行的独立任务将处理一项工作请求，这个请求是由在执行的RTQEL来代表的。该独立任务的所有其它工作请求都放在RTQEL中进行排队。这个排队受优先级的指挥，在等优先级的情况下，则为先进先出（FIFO）。当任务处理完RTQEL的时候，RTQEL排队中的第一名RTQEL就开始执行并送主任务。如果没有排队的请求（RTQEL）了，那么任务就处在休止状态并在系统中等候新工作的到来。独立任务的所有

工作请求能被任意地放置在排队管理控制之下，这种控制直接将每一 RTQ EL 列入“实时排队” (RTQ) 内。每一 RTQ 均由使用者去指令产生，该指令规定排队有五点特征：

- 它的专用名称，使 RTQ 一致
- 它的长度，即过满之前 RTQ 中掌握的 RTQ EL 的最大数目
- RTQ EL 从 RTQ 中取出并送给独立任务进行处理的次序 (调度优先权，FIFO，LIFO)。
- 关于过满的安排：RTQ EL 从 RTQ 中取出，但如果排队过满则停止 (最新，最老，最低优先权的 RTQ EL)。
- RTQ 能否流畅地把 RTQ EL 送给独立任务 (能或不能)。

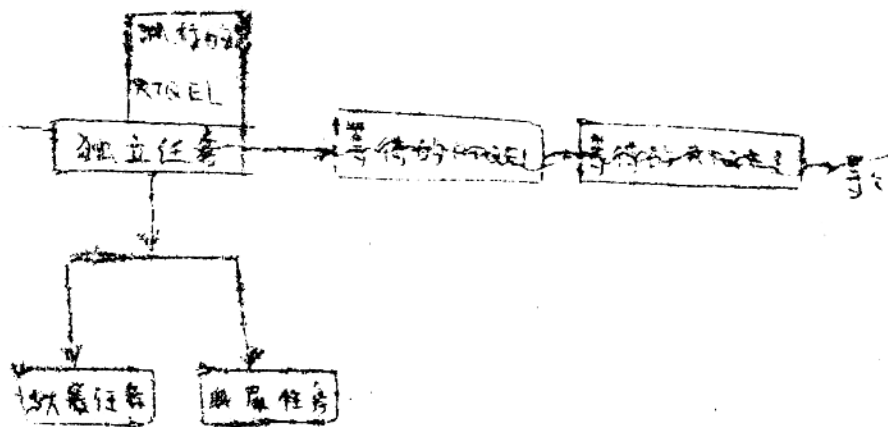


图 6—独立任务及 RTQ EL

图 7，实时排队单元控制，是“RTQ EL 受 RTQ 控制”这样的逻辑结构的一例。属于 RTQ 的五点特征和其它控制信息都被实时



序列控制块 (RTQCB) 所掌握。在此例中, 如果他们不受 RTQ 控制的话, RTQEL 送给独立任务时, 应按照一、二、三、四的次序, 这就是它们有关的调度优先级的顺序。然而, RTQ 有先进先出 (FIFO) 的指挥特征, 所以 RTQEL 送给任务将按三、一、二、四的次序。

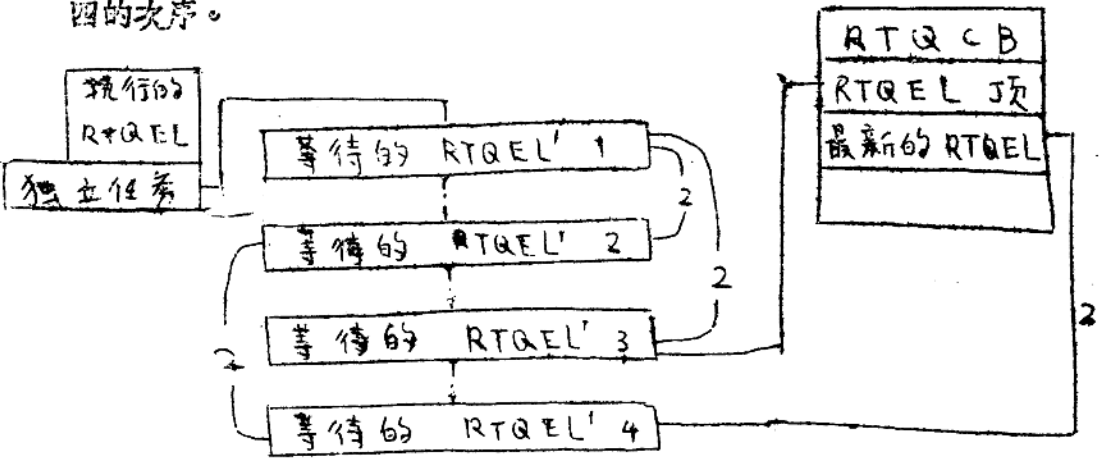


图 7—实时序列单元控制

如果不采用排队管理的话, 那么排队独立任务的 RTQEL 就要无限制地堆积起来, 除非任务处理“工作请求”的速度比产生“工作请求”的速度快一些才行。排队管理就是对排队的请求进行附加控制, 其控制的方法是限制为独立任务掌握的 RTQEL 的最大数目。这一点也可以用于间接控制系统的负载, 其方法是在一项独立任务的处理工作还没有完成之前, 不把工作交给另一项独立任务。

RTQ 的数目及其结构完全由使用者来确定。一项 RTQ 可以包