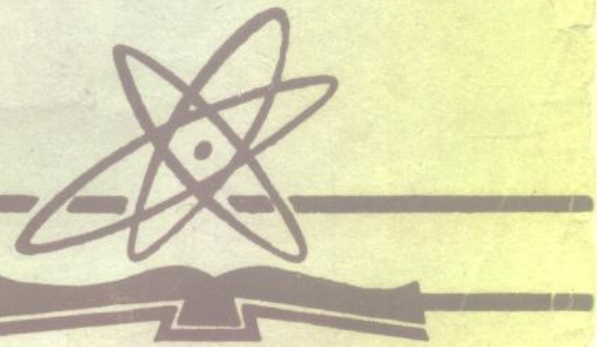
操作系统原理

成都电讯工程学院

汤子瀛 杨成忠 编



国防工业出版社

操作系统原理

成都电讯工程学院

汤子瀛　杨成忠　编

国防工业出版社

内 容 简 介

本书专门介绍计算机系统的一个重要软件——操作系统。全书共分八章，第一章概括地介绍了操作系统的基本内容和概念；第二章较系统地阐述了进程的概念，通讯和调度方法；第三、四、五和六章介绍操作系统中各种资源的管理；第七章叙述了操作系统的结构设计；第八章则通过一个实例RTOS来说明操作系统的技术实现方法。

本书可作为计算机专业的教科书，也可作为从事计算机工作的科技人员，学习操作系统的一本参考书籍。

操作系统原理

成都电讯工程学院

汤子瀛 杨成忠 编

国防工业出版社出版

北京市书刊出版业营业许可证出字第074号

西北电讯工程学院印刷厂印刷

内 部 发 行

开本 787×1092 1/16 印张 19 7/8

印刷字数 507 千字 印数 1—18000 册

1981 年第一版 1981 年 2 月第一次印刷

统一书号：N15034(教-68) 定价：2.06 元

前　言

随着计算机技术的迅猛发展，计算机系统的硬件结构日趋复杂，拥有的资源也愈来愈多。为提高这些资源的利用率、组织好计算机的工作流程，从而增强系统的处理能力，以及使用户能方便和有效地使用计算机，在计算机系统中增设了操作系统。

操作系统是用以实现上述功能的程序的集合，属于系统软件。目前操作系统不仅配置在大型计算机系统上，而且也普遍用于中，小型计算机中。例如，国产的 DTS-130 机上，就配置了实时磁盘操作系统 RDOS 和实时操作系统 RTOS 。为了适应这一情况，在国内计算机专业中也都相应地设置了操作系统这一课程。

本书共分八章。第一章是通过对批量处理系统、分时系统和实时系统的介绍，使读者对操作系统的基本内容和概念有一概括了解；第二章较系统地阐述了进程的基本概念，进程的控制、通讯和进程的调度方法以及死锁等问题；第三章对作业的组织、控制以及作业的管理和调度方法作了较详细地介绍；第四章系统地介绍了存贮管理的基本概念，各种存贮管理方法，对分区分配、分页存贮管理、分段存贮管理作了详细介绍；第五章对通道，缓冲技术，设备分配程序和处理程序作了较详细的介绍；第六章较详细地阐述了文件的组织，文件目录，存取控制方式以及文件的使用等，第七章对设计操作系统常用的模块接口法，有序分层法以及基于管程和类程概念的设计方法作了初步地介绍；第八章是通过 DJS-130的RTOS 来说明一个简单操作系统的具体技术实现方法。

本书是按 100 学时编写的，若按 80 学时讲授，建议免讲带星号的部分。

本书的第 1,2,5,7,8 章由汤子瀛编写，3,4,6章由杨成忠编写。参加本书审稿的单位有：重庆大学(主审)，清华大学，上海交通大学，国防科技大学，华中工学院，黑龙江大学，北京工业大学，北京计算机三厂，东北重型机械学院等。与会代表在审稿过程中提出了许多宝贵意见，本书在编写过程中还得到许多同志的帮助，特在此表示感谢。由于编者水平有限，书中错误和不妥之处在所难免，恳请读者批评指正。

编者　1980 年 8 月

目　录

第一章 操作系统引论

现代计算机系统通常拥有数量可观的硬件和软件资源，前者是指中央处理机、存贮器和I/O设备等物理资源，后者包括程序和数据。为提高这些资源的利用率和增强系统的处理能力，而出现了监督程序。到六十年代中期，监督程序又进一步发展形成了操作系统。所谓操作系统系指，用以控制和管理硬件和软件资源，以及方便用户的程序的集合。因此，我们可以从操作系统是资源管理程序这样的观点来研究操作系统，而形成所谓的资源管理观点。由于通常把系统中的资源分为四类：处理机、存贮器、I/O设备和信息(程序和数据等)，相应的操作系统就应包括这样几个部分：(1)用于控制和管理处理机的程序；(2)控制和管理存贮器的程序；(3)控制和管理I/O设备的程序；(4)控制和管理程序和数据的程序。由此可见，操作系统是计算机系统中极其重要而又基本的系统软件之一。

本章是从用户观点对操作系统作一粗浅的介绍。它通过对批量处理系统、多道程序系统、分时系统和实时系统等的介绍，使读者了解操作系统是如何产生和发展起来的，各种类型的操作系统中所要解决的主要矛盾是什么？它们又是如何解决这些矛盾的？并对操作系统中的一些基本概念有一初步了解。

1.1 批量处理系统

1.1.1 作业流、作业和作业步

通常，一个用户程序是有着一定独立性的程序段的有序集合。这些程序段一般都是以某种语言形式写出，例如用ALGOL、FORTRAN、或汇编语言等。在一个用户程序中各程序段可用不同的语言。然而，机器在执行时只能识别机器语言，因此每个程序段在执行前，都必须首先被翻译成机器语言。为使机器知道所用的语言形式，又必须以某种命令形式（作业控制卡或作业说明书）告诉系统，以便系统调用相应的翻译程序将之翻译为机器语言。此外，当用户程序开始时同样需要以某种命令形式告诉系统。用户程序及其所需的数据和命令一起形成一个作业（job）。

逻辑上作业是由有序的作业步组成，每个作业步又是由完成作业中某一相对独立事件的程序和数据构成，并由命令定义之。例如，一个作业可由“翻译FORTRAN源程序为机器语言”，“装入编译出的程序”，“执行该目标程序“三个命令所定义的三个作业步组成。

在批量处理环境中，通常都是把一批作业有序地排在一起形成一个作业流，这同样也采取命令形式来标识一个新作业的开始和前一作业的结束。

1.1.2 批量处理系统的引入

在五十年代，计算机硬件大都由电子管构成，其体积庞大，内存容量小且速度慢，每秒钟只能运行几千次。软件也处于低级发展阶段，仅有汇编语言以及少量的服务性程序，计算机主要用于科学计算。

此时计算机的工作基本上采用人工操作，它将纸带（卡片）装进纸带输入机（卡片输入机），把程序和数据输入，然后通过控制台开关启动程序。在程序运行中也是利用控制台开关对它进行控制。仅当程序完成并取走它的纸带和计算结果后，才让下一个用户上机操作。这种人工操作方式具有如下特点：

1. 用户独占全机：一个计算机为一个用户所独占，其全部资源由他一人支配，因此用户可以较方便地使用资源，不会出现因资源为别的用户占用而等待的现象，但资源利用率却非常低，因为很多小型题目无须利用计算机所提供的全部资源。

2. CPU 等待人工操作：用户所使用的纸带等，都只有在他上机时才装入相应的设备，在此期间机器空闲。同样，当计算完成后在做卸带取卡等人工操作时，机器也无事可做，以致机器利用极不充分，特别是在运行短程序时更为突出。

以上两点充分说明人工操作方式严重地损害了资源的利用率，此即所谓的人机矛盾。对第一代计算机来说，人机矛盾尚不突出，因为计算机本身所拥有的资源并不多，内存容量一般只有几 K，I/O 设备也只有几件。此外，由于计算速度低，计算所需时间相对较长，因此人工操作时间所占比例也不算太大。但随着计算机规模的不断扩大，例如，内存容量由几 K增至几十K以至几百 K，I/O 设备由几件增至几十件以上，计算机的速度由每秒几千次增到百万次，人机矛盾就变得严重起来，如人工操作方式再不改变，资源利用率可能降为百分之几甚至更低。

此外，随着 CPU 速度的大幅度提高，使得 CPU 和 I/O 设备间速度不匹配的矛盾更为突出。为了解决速度不匹配的矛盾，并使它们之间能重叠操作，又出现了一个关键性硬件——通道设备。而缓冲技术特别是脱机输入输出操作，又进一步缓和了 CPU 和 I/O 设备间速度的矛盾。早期的脱机输入操作是把用户程序和数据在一台外围计算机的控制下，预先从低速输入设备输入到磁带上，当 CPU 需要时就可直接从磁带输入至内存。在脱机输出时，CPU 只要把结果送至磁带上，然后再在另一台外围机的控制下，使之由相应的输出设备输出，这样就大大加速了输入输出过程。

在脱机输入输出方式中，由于事先已把若干个作业记录在一盘磁带上，这意味着用户程序的处理顺序已经排定。机器将首先处理磁带上的第一个作业，即把它由磁带传送至内存后启动它，并把控制权交给作业 1。当作业 1 完成后又把控制权交还给系统，系统再把磁带上的下一个作业输入内存并启动它，把控制权交给第二个作业。按这种方式对磁带上的作业自动地一个接一个地进行处理，于是便形成了批量处理系统。不难看出，单道批量处理系统是在解决人机矛盾和 CPU—I/O 速度矛盾的过程中，也就是在提高资源利用率的过程中发展起来的。此时，为对系统中的作业和计算机各种资源进行监督和管理，又相应的配置了监督程序（Monitor）或管理程序（Supervisor），它不仅是操作系统的前身，而且也是操作系统的核心部分。

1.1.3 批量处理监督程序的功能和组成

早期批量处理系统中的操作组织如图 1-1 所示，监督程序放在内存的顶部或底部。包括编译程序、汇编程序、装入程序等的系统程序记录在磁带上，被称为系统带。作业流由卡片机输入，源程序被翻译为目标程序后暂存在磁带上，被称为目标带。

该批量处理系统的主要目的是使整个作业流能自动地、顺序地运行，以节省人工操作时

间和改善机器的利用情况。当作业都是短小任务时，批量处理能获得良好的效果。在最基本的批量处理形式中的工作流程如图 1-2 所示。

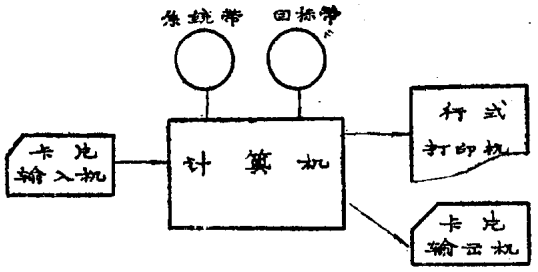


图 1-1　早期批量处理系统的组织

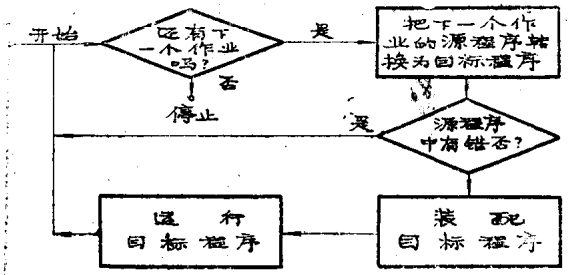


图 1-2　简单批量处理流程图

为使监督程序能代替操作员对作业进行控制和管理，用户必须通过某种命令方式，把与作业有关的信息提供给系统。系统愈完善，能处理的事情愈多，需要的信息也愈多。当系统接到这些命令后，首先对该命令作正确性检查，若无误，再进一步通过查找一张命令表，而转至相应的命令处理程序，如编辑程序。在该命令表中包括了所有的命令，在每个命令表目中记录有命令标识符及该命令程序入口在内存或外存中的地址。这样，当系统接到命令后便到该表中去查找。若查到，便可转至相应的命令处理程序；若在该表中查不到发来的命令，则认为该命令非法。为实现这些功能，系统中必须配置相应的命令解释程序 (Command interpreter) 。

在批量处理环境下，系统还必须能根据命令给出的信息，识别一个作业或作业步的开始和结束，以便当一个作业或作业步完成时，无须操作员的干预便可自动地过渡到下一个作业或作业步。这就需要有一个作业定序程序 (Job Sequencer) ，与命令解释程序相配合来完成这一任务。在定序程序中利用了一张作业描述表来记录作业标识符、估计运行时间和有关作业的某些限制（最大运行时间，打印行数）以及作业的累计费用等。当一个作业完成时，定序程序便检索该表，从中找出下一个作业，把它从外存调入内存投入运行。由命令解释程序和定序程序两部分，组成了监督程序中的作业控制部分。

由于内存容量有限，因此常把一些系统程序和用户程序放在后援存贮器中。例如，当系统接到一条编译源程序的命令时，便首先从后援存贮器中把相应的编译程序调入内存，为此，还必须事先为编译程序找到一个内存空间来存放它。由于通常在内存中并没有足够容纳编译程序的空白空间，因此必须将原来内存中的其它程序调至后援存贮器，以获得足够大的空白空间。为此系统必须知道被编译源程序的所在位置，记住被编译为机器语言的目标程序所在地址，以备需要时查找。此外，虽然源程序已翻译成目标程序，但它只是单个程序的集合，尚待把它们装配成一个完整的用户目标程序。同样，系统也应首先找到内存空间来存放它，然后才能开始执行。由此可知，监督程序需要具备内存分配的功能。内存分配是借助于一张内存使用表来实现的，表中记录了哪些内存区域已经使用，哪些是空白空间。当一个作业到达时可通过查找内存使用表，为该作业找到空闲的内存空间。

用户程序和它所需要的数据以及相应的控制命令，都需要通过输入机输入。当作业执行后，又要求把所得的结果以打印或穿孔的形式输出。为使操作员了解作业进展情况，每当作业和作业步在开始或结束时，都应打印出相应的信息通知操作员。此外，虽然在正常情况下无须人工干预，但在异常情况下仍须人工干预。这就须要把作业在执行中的有关信息，状态

通知操作员。因此，读写控制是监督程序应具备的基本功能。

当一个用户希望把它所得到的结果作为永久性或暂时性记录保存在系统中时，系统便将该结果信息组成文件，存放在磁带或磁盘上。文件通常被认为是有关联的数据记录的集合，记录则是有序字符的组合。例如，文件可以是用户的源程序或某单位职工的名册。在这种情况下，甚至各种低速I/O设备，例如纸带输入机，打印机等，也都可看做是按字符流形式存取的文件。因此，无论是文件或I/O设备都可采取同一方式，用符号名称进行访问。文件管理的主要目的是，为用户提供一种存贮信息和检索信息的手段。为了记录每个外存的使用情况，系统为每个外存准备了一张使用表。另外，系统中还保存了一张记录每个文件所在地址的文件目录。这样，当用户需要检索一个文件时，系统便可查阅文件目录，从而迅速把文件调出。事实上，文件管理的方法极像图书馆的管理情况。

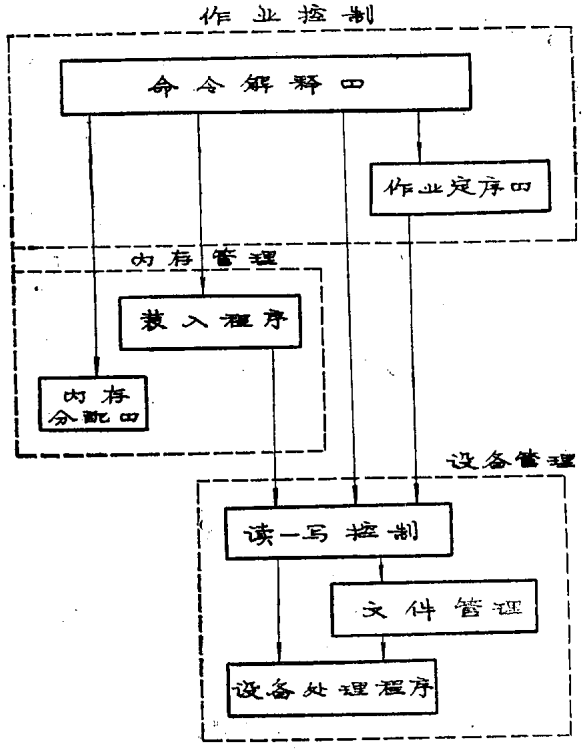


图 1-3 监督程序的粗框图

在系统中具体的I/O操作由I/O设备管理程序中的第三部分——设备处理程序来实现，这些操作包括启动指定的I/O设备。处理来自I/O设备的中断等。为协助设备处理程序来完成具体I/O操作而设置了一张设备控制表DCT，它记录了每个I/O设备的标识符、现行状态（忙闲），连接的控制器等。

综上所述不难看出，无论是作业控制，内存管理，还是设备管理都是借助于一张表格来实现的，它们是监督程序实施管理功能的数据基础，故通常称之为数据基(Data base)。图1-3示出了监督程序的粗框图

1.2 多道程序系统

1.2.1 多道程序的引入

虽然批量处理系统已大大减少了人工操作的时间，提高了机器的利用率。在理想情况下，操作员所做的全部工作只是，把所有作业适当地分批，装入相应的输入机，启动机器，于是作业便一个接着一个地进行，最后把所得的结果取走。即使如此，对于某些作业在发出I/O请求后，还必须等待，而在单道程序运行中的作业，等待就意味着机器空闲，特别是由于I/O设备的低速性，将导致机器的利用率非常低。图1-4示出了单道程序运行情况。在t_1时刻，用户程序发出I/O请求后进入监督程序，作适当处理后在t_2时刻启动I/O设备工作，用户程序处于等待I/O操作完成状态，在t_3时刻I/O完成，发出结束中断信号后进入

监督程序，t_4 时刻又恢复用户程序运行。（图中 $t_2 \sim t_3$，$t_6 \sim t_7$ 为 CPU 空闲时间）

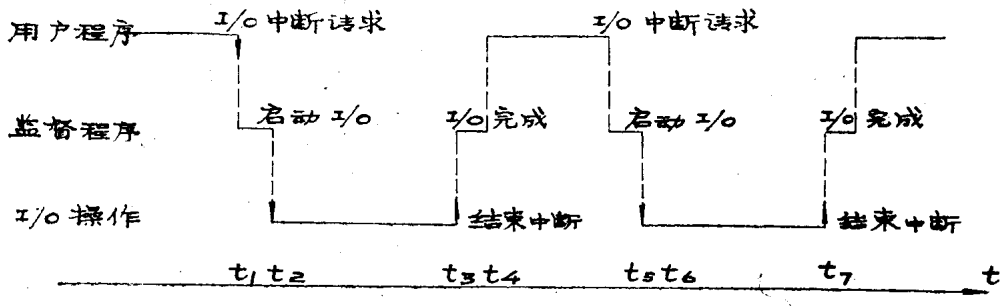


图 1-4　单道程序运行情况

为了改善 CPU 的利用率而引入了多道程序，即同时把几个作业放入内存，它们分时共用一台计算机。CPU 先对第一道作业进行处理，当它需要 I/O 时，CPU 处理了它的 I/O 请求后便转向第二道程序，使第一道程序的 I/O 操作和第二道程序的处理并行。当第二道程序需要 I/O 时，CPU 处理了它的 I/O 请求后又转向第三道程序，使第三道程序的处理与第一、二道程序的 I/O 操作并行。图 1-5 示出了两道程序的运行情况（其中 $t_4 \sim t_5$、$t_8 \sim t_9$ 为 CPU 空闲时间）。

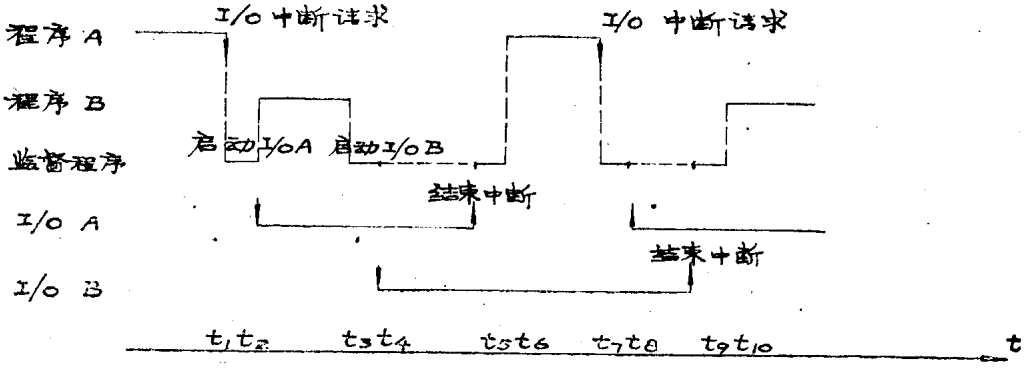


图 1-5　两道程序运行情况

随着计算机的发展，具有各种用途的 I/O 设备也愈来愈多，在单道程序运行时，很难充分利用所有的 I/O 设备。此外，内存容量也在不断扩大，以适应日益膨胀的大型作业的需要。但作业并非总是大型的，经常遇见的倒是中、小型作业，这样，就会有很多内存单元未能装入有用信息而白白浪费。

在批量处理系统中，运行情况往往不能像上述那样理想，即作业一个接一个地往下运行而无须人工干预。实际上，用户程序经常出现这样或那样的错误，如“越界”、“地址错”、“阶上溢”等。一旦出现了这些错误，程序便无法运行下去，须经操作系统分析原因后，或从头做起（复执），或暂停下来等待人工干预，这又意味着处理机时间的损失。但在多道程序环境下，当某程序出错时，系统把现场记录下来后便马上转向另一道程序继续运行，而不致严重影响系统效率。

在现代计算机中广泛采用人机会话。所谓人机会话，是指用户通过终端设备，如控制台、打字机，向机器发布各种命令，系统通过用户终端设备向用户回答或提出问题，这无论对新

程序的调整，还是对计算机的辅助设计都带来极大方便。但人机会话的速度很低，当人用控制台打字机向机器发布命令时，每秒钟只能发送几个字符，仅在一条命令送完后机器才予以响应。因此，在单道程序运行时机器效率受到严重影响。然而在多道程序环境下，命令的输入是穿插在其它程序中进行的。宏观地看，机器一面运行程序，一面做缓慢的人机会话，这就不至于严重地影响机器效率。

1.2.2 多道程序系统中管理程序的功能

由上所述可知，多道程序系统是利用CPU的等待时间来运行其它程序，这就显著地提高了资源的利用率，从而增强了系统的吞吐能力。因此，现代的大、中、小型计算机广泛地采用多道程序设计。但它也出现了一些新的问题，只有将这些问题妥善地解决好，才能实现有效地多道程序运行。其中的基本问题是，用户提交给系统的作业所需要的资源总和，远比系统所拥有的资源多，使得系统无法同时满足所有作业的资源要求而将它们开动起来。于是，这些作业必将争夺资源，以便投入运行。为使多道程序能有条不紊地运行，系统中必须增设管理程序，以便把这些资源管理起来，遵循一定的策略把这些资源分配给某些作业。按照系统资源的类型可把管理程序分为四种：

1. 存贮器管理

在单道程序运行时，系统的全部资源为正在执行的作业所独占。然而在多道程序运行时，由于内存中同时装有若干道程序，故内存管理的第一个功能是，按一定策略进行内存分配，使它们不致因相互重叠而丢失信息。此外，还必须防止因某道程序出现异常情况而破坏其它程序。当然，所有的用户程序更不能破坏管理程序，这就要求存贮管理提供存贮区保护的功能。

当一个作业所要求的内存容量或几个作业所要求的内存容量总和，超过实际内存容量时，为使这些作业都能投入运行，内存管理的第三点是必须具有内存扩充的能力，亦即系统中应采用虚拟存贮器技术。所谓虚拟存贮器技术，是指通过某种机构，把由内存和后缓存贮器组成的两级存贮器系统，变为可由用户程序直接存取的一级存贮器系统。通常由于磁盘的容量远大于内存容量，因此用户所看到的内存容量就比物理内存大。这就使得一个大的用户程序可以在这样的系统中运行，该系统所具有的物理内存比实际需要小得多。

2. 处理机管理

在单处理机系统中作单道程序运行时，处理机为一道程序所独占，而在多道程序运行时，处理机则为几道程序所共享。这样必将引起各道程序对处理机的争夺。因此，处理机管理的基本功能是，按照一定的策略（如具有最高优先数的程序优先处理）把处理机分配给某个程序，使之运行一指定时间（例如500ms，该时间称为时间片），或直到该程序处理完毕，然后再分配给另一用户程序，这样几道程序便可有条不紊地在一个系统中运行。

3. I/O设备管理

在多道程序运行时，同样也会发生几道程序对I/O设备的争夺。因此，设备管理的第一个功能也是按照设备类型和一定的策略（如优先数策略）把I/O设备分配给某些作业，与此同时还应分配相应的通道和控制器。当该作业不再需要时便予以收回。设备管理的第二个功能，是启动指定的I/O设备进行数据传送操作，和对通道发来的中断请求及时作出响应和处理。

4. 文件系统

在较为完善的操作系统中，总是把大量有意义的信息以文件形式存放在各种存贮器中，以提供给所有的或指定的用户使用。系统还允许用户把它所输入的信息或计算所得之结果保存在系统中，便于自己或有关用户今后使用。如果在这样的系统中未配置相应的文件系统，用户欲存取各种存贮器上的信息就相当繁琐，在多道程序环境时更不可想象。为此，在现代操作系统中都必须配置有文件系统，以提供相应的存贮和检索这些信息的手段。

1.3 分 时 系 统

1.3.1 引 言

1.3.1.1 分时系统的引入

单道批量处理系统卓有成效地提高了机器的利用率，引入多道程序技术又使之得到显著改善，从而可获得比较理想的机器利用率。但这样的系统在下述情况下仍不敷用户需要，而有待改善。

1. 用户一旦把其作业提交给系统后，便失去自己对作业的控制和修改能力。因此它必须以作业说明书的形式向系统提供控制信息，还须仔细考虑运行过程中可能出现的情况，把处理方法告诉系统。这就给某些用户增加了困难，例如新程序的调试者，由于他无法预测可能出现的所有事件，因此他强烈地希望系统能边运行他的程序，边告知处理情况，以便及时修改其程序和参数。即系统应具有用户与系统以及用户与自己的作业交互作用的能力。

2. 在批量处理系统中用户提交作业后，通常要经几小时甚至几天的延迟才能得到所需要的结果，这样长的周转时间对那些仅计算一个很小题目，或只对已存文件作几行修改的短作业用户很不利，因此应该尽量缩短周转时间，例如缩短到几分钟，以满足短作业用户的要求。

3. 对上述系统而言，尽管用户要解的题目很小，也必须自己把程序和数据送至机房或计算中心。这对本单位甚至本地没有机器的用户是极不方便的。倘若用户能通过自己的终端设备，直接将所用程序和数据发至计算中心，将是最理想的。为此，要求系统具有能与远地用户终端通讯的能力。基于上述需要，在批量处理系统的基础上，引入远地作业进入方式和人——机交互作用而形成了分时系统。

1.3.1.2 分时概念与分时系统

并行操作这一概念早已为大家所熟知，它是为提高资源利用率而实施的一种技术。其中CPU和通道并行操作，通道与通道并行操作，通道与I/O设备并行操作已成为现代计算机系统的基本特征。为了节省设备，CPU、通道和I/O设备之间的并行操作又按分时方式共享系统资源。与三种并行操作相应的有三种分时：CPU与通道分时使用内存、只读存贮器、数据通路等；通道与通道分时使用CPU、内存、通道的公用控制部分等；同一通道中的I/O设备又分时使用内存、通道等。

分时概念并不局限于上述三种分时，在多道程序环境中，分时概念又有进一步的扩充，而形成多道程序分时共享硬件和软件资源。如果每道程序一次运行一个时间片，且都经由用户终端与一个用户相连，用户便可通过终端与之交互作用。虽然该计算机系统是由若干用户共

享，但他们彼此并不感觉到有别的用户存在，而好像整个系统为他所独占，这样的系统称为分时系统或多路系统。不难看出，分时系统的基本特征为：（1）同时性：若干个用户能同时或基本上同时使用计算机系统；（2）独立性：用户可以彼此独立地操作，而不会发生相互混淆或破坏现象；（3）及时性：用户能在很短的时间内获得对系统所提要求的回答；（4）交互作用性：用户能与系统进行人—机对话。

1.3.1.3 分时系统的优点

1. 促进了计算机的普遍应用　仅在分时系统出现后，计算机才真正获得普遍应用，这是因为，用户仅需配备一台价格远比计算机便宜的终端设备，就可方便地使用计算机系统。例如，一个单位仅需一个分时系统，便可供本单位的几十个科、室在自己的办公室，通过终端设备使用该计算机系统；学校中有此系统便可为成十组的学生同时开出上机实验。分时系统进一步扩大为远程分时系统后，可为不具备计算机的多个远地用户服务。

2. 节省开支、减少维护人员　用户在分时系统中不必自备计算机、这样不仅节省了开支，而且也减少了大量的维护人员。事实上，就我国目前情况而言，自备计算机必须相应地配备专职维护人员。显然，这种情况必然使得人和机器都不能充分发挥作用。

3. 可作为工程设计和方案论证的得力工具　很多科研单位和工业部门在进行工程设计时，往往借助于数学模拟方法。若使用分时系统，设计者就可通过与系统的交互作用，既方便又及时地发现设计中的问题，确定所需参数及最佳工作状态。

4. 显著地提高了研究、检查和调整程序的效率　即使一个经验丰富的程序设计员，在编制一个不太复杂的程序时，也难免要出现错误。如果这些初编制出的程序在批量处理系统中调试，要经过很久的延迟才能获得一个没有运行完的结果，须经修改后再试。因此调整一个程序要花费很多时间。然而在分时系统中，可以像早期程序员使用计算机那样，边调整、边思考，边修改，又不会影响机器利用率。这就大大缩短了程序调整的周期。在这种服务方式中，用户也无须担心程序设计过程中的小疏忽或错误。一发现有错便可及时纠正，不致因程序中的小错误而被强迫撤离系统。

5. 进一步充分利用系统资源　一个大型计算机系统，特别是具有巨型机和昂贵专用设备的系统，在一个单位甚至一个地区都难于充分发挥其效能。通过大量的远地用户分时使用，便可大大提高系统硬件资源和软件资源的利用率。

1.3.2 实现分时的方法

1.3.2.1 远地作业进入(remote job entry)

远地作业进入是由批量处理系统走向分时系统的第一步。它是在批量处理系统的基础上，增加一个远地作业程序，它专门处理由远地终端发来的作业，将它们装入后援存贮器中，如图 1-6 所示。当终端用户要求先编译自己的源程序然后再执行它时，可通过纸带或卡片输入机将自己的作业送入系统，或者直接用打字机打入。系统中的远地作业程序将它复制到后援存贮器中。当作业输完后，系统便按一定的算法将它插入相应的作业队列中形成批量作业流。以后系统便按批量处理方式对它们进行处理，并把所得结果送入后援存贮器中，最后再从后缓存贮器取出，送入该作业的用户终端机把它打印出来。这种系统实际上仍然属于批量

处理系统、从任何意义上说它都不具有交互作用的能力。但它对远地用户不论是在方便性，还是在响应时间上都会带来好处。

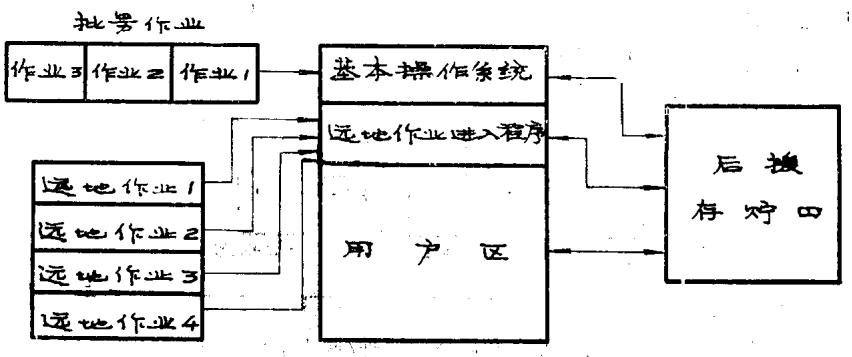


图 1-6　远地作业进入示意图

远地作业进入的进一步发展，是允许用户与系统中的某些标准子程序交互作用，如创建一个文件、编制一个已存文件等。这对程序的编制、调试都是方便的。但只要作业一旦由批量处理系统进行处理时，就一直运行到结束，终端用户就不能再和它交互作用。因此它还不是一个完全的分时系统，只能说它是由批量处理系统向分时系统迈进了一步。

1.3.2.2　调进/调出（roll-in/roll-out）

为了实现完全的分时，使终端用户不仅在编辑时能和系统进行会话，而且在作业处理过程中的各个阶段也都能与自己的作业交互作用。这就要求系统能及时响应用户的请求，而不能象批量处理系统那样：一旦一个作业投入运行，便一直运行到结束，然后再调入第二个作业运行。在最简单的分时系统中，内存中只能存放一道程序做为现行作业，其它作业都放在后援存贮器上。为使系统能及时响应用户请求，每次现行作业运行一个时间后（例如 500ms）便暂停该作业的运行并把它从内存调至后援存贮器（调出），再从后援存贮器中挑选一个作业装入内存（调进）。作为下一个时间片的现行作业投入运行。如果在不太长的时间内，例如 3 秒，能对每个作业轮流地运行一个时间片，那么每个用户都会认为系统对自己的作业做出了及时的响应，从而实现了终端用户与自己的作业交互作用。但在这种交互方式中，很大一部分时间都花在内存与外存的对换上，如图1-7所示。显然，在此期间 CPU未被有效地利用。

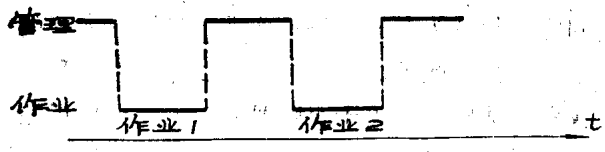


图 1-7　简单的调进/调出工作情况

为了改善 CPU 的利用率，又引入了所谓“前台”(Foreground)和“后台”(Background)的概念，相应地把内存也分为“前台”和“后台”两部分。“前台”存放按时间片调进/调出的作业流，工作方式同前。“后台”部分存放批量处理作业，仅当“前台”调进/调出时，才处理“后台”的批量处理作业，如图 1-8 所示。它既不影响对终端的响应时间，又改善了机器的利用率。

在“前台”只有一个作业流的前后台系统中，虽然已改善了 CPU的利用率，但仍有潜力可挖。例如，当前台要进行内存与外存的对换时，后台作业也正因 I/O 请求而处于等待状态，CPU 又被空闲起来。此外，它对系统的分时能力并未带来好处。为进一步提高机器利