目　录

第一章 概 述

1.1 为什么要有操作系统

1.1.1 什么是系统

所谓“系统”，是指为了达到某种目的而把相互间具有有机联系的各部件结合起来所组成的整体。“系统”这一概念，在我们的周围大量存在着。一般来说，一个系统有如下特点：

(一)有特定的目的——设置一个系统是要该系统完成一定的任务，它的存在具有明确的目的性。

(二)系统由多个成分所组成——每个系统，必须由许多子系统构成，这些子系统就是系统的组成成分。系统完成的任务不同，组成系统的成分亦各不相同。

(三)各组成成分相互联系，完成各自特定的功能——组成系统的各子系统，具有各自的功能，为了实现整个系统的功能，各组成成分之间必须相互协调和配合。

(四)有输入、有输出。

(五)系统在一定的环境下工作。

毫无例外，操作系统也具有一般系统的上述特点，也是为一定的目的而设置的，它也由相互联系的、具有各自特定功能的若干成分所构成。

在今天的计算机系统中，从巨型机到小型的个人计算机，都配置了相应的操作系统。然而，早期的计算机，是没有配置操作系统的。

1.1.2 早期的计算机系统

计算机系统通常由中央处理机（*CPU*）、主存储器和外部设备（一般把中央处理机和主存以外的设备，统称为外部设备）三个基本部件所组成。早期的计算机系统以*CPU*为中心，这三个部件是按照图1-1的方式联接的。

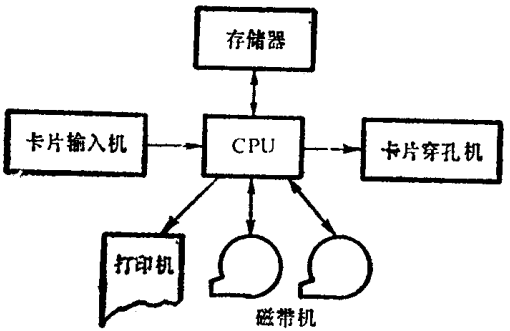

图 1-1　早期的计算机系统

早期的计算机系统，不配有任何有助于用户的程序，它只给用户提供指令系统的原始功能，故常称之为裸机。当时，人们是通过操纵计算机上一系列复杂的开关来使用计算机的，这样，使用计算机便成为少数计算机行家的事了。

早期的计算机系统，是一个单用户独占的单道程序系统，系统中的各设备是串行工作的，我们不妨把整个计算机系统看作一支乐队，把各部件好比各种乐器，这些乐器只能挨个地表演独奏（串行工作），这支乐队的指挥，则相当于系统中运行的那道程序。若运行的是一道算题程序，当该程序在进行计算时，各台*I/O*设备是停止工作的；当该程序需要输入输出时，*CPU*必须停止工作而等待输入输出完毕。这种工作方式存在以下问题：当一个用户程序在运行期间，系统的全部资源为该用户程序所独占。另外，由于设备的串行工作，使得在*CPU*执行计算时，任一台 *I/O* 设备不能工作而处于空闲状态，而且，任一台 *I/O* 设备工作时，系统中其余各台 *I/O*

设备也均不能工作，故 *I/O* 设备的利用率极低。同时，当任一台 *I/O* 设备工作时，*CPU*必须等待输入/输出的完成而出现空等现象，使*CPU*的利用率也极低。例如，有两个用户要求运行他们各自的作业*A*和*B*，若作业*A*先运行，*A*运行0.2秒后，需用0.6秒的时间将运行结果送入磁盘，待作业*A*运行完毕，接着由作业*B*运行，它也运行0.2秒，然后用1秒的时间把结果打印输出，如图1-2所示。可见，完成*A*、*B*两个作业，共需2秒，然而*CPU*却只工作了0.4秒，其余1.6秒都处于空等状态，*CPU*的利用率只有20%。我们知道，*I/O*设备的速度比中央处理机的速度慢得多，例如，从纸带输入机上输入一个字符，要花1 ms时间，而*CPU*接收这个字符，并把它存放好，可能只要花10μs或更少的时间，因此，在1 ms时间里，中央处理机只工作了10μs，占整个时间的1%，*CPU*有99%的时间都处于空等状态。尤其在电子技术发展很快的今天，*CPU*运算速度不断提高、外设的种类和数量愈来愈多，各种*I/O*设备与*CPU*之间在速度上不匹配的矛盾更为明显，更加不能容忍了。

如何提高中央处理机与外部设备的效率呢？解决办法之一是采用中断系统。例如，要输出打印一批数据时，*CPU*首先把要输出的数据存放在一个缓冲区中，接着通过执行一条输出命令，将缓冲区中的第一个数据送到输出接口，并通知打印机："有数据，请输

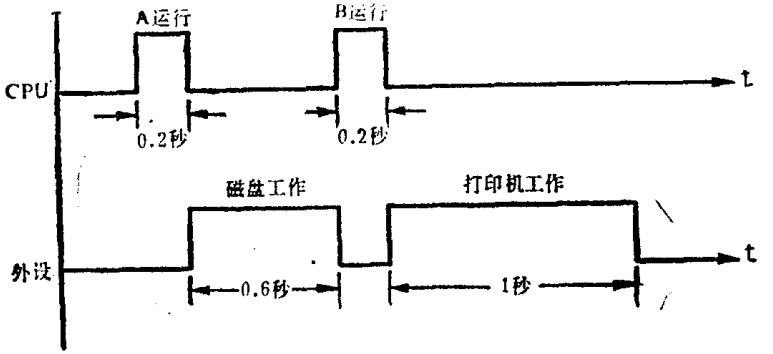

图1-2 *CPU*与外部设备串行工作

出”，然后，主机便可以去做其它的工作，而打印机接到通知后，便作好准备，并向主机发回一个信号，表示“已准备好，可以输出”。于是将数据线上的数据送入打印机缓冲寄存器中进行输出打印，然后向*CPU*发出中断，请求打印下一个数据，此时，*CPU*暂停原来的工作，将缓冲区中的下一个字符交给打印机后，便返回到被中断处继续它原来的工作，而不空等了。当打印机打印完一个数据后，又产生中断，如此重复，直到把缓冲区中的数据全部打印完为止。

中断引入后，虽然使主机和外部设备可以重叠工作，提高了*CPU*的利用率，但是，*I/O*操作仍由*CPU*来控制，*I/O*设备每输入或输出一个字符，都要产生中断而打扰*CPU*的工作，故*CPU*的利用率提高有限。例如，要传输1000个字符，外部设备要向*CPU*发1000次中断，相应地，*CPU*要作1000次中断处理，然而，其中只有最后一次中断处理，才是作传输结束的处理工作，其余999次，都是用于传输下一个字符。可见，以*CPU*为中心的计算机体系结构的效率低，速度慢，而且人机联系既直接又密切。若能做到在*I/O*设备把这1000个字符传送完毕后，才向*CPU*发中断，那么，*CPU*仅在一批信息传输结束时，才作结束处理，这将会进一步提高*CPU*的效率。

1.1.3 具有I/O通道的计算机系统

由于*I/O*设备与*CPU*的工作速度相差悬殊，出现了*CPU*等待*I/O*传输的现象，虽然引入了中断，收效并不显著。为了使*CPU*摆脱忙于应付*I/O*中断处理的局面，进一步提高计算机系统的并行性，在硬件上引进了*I/O*“通道”部件。

*I/O*通道又叫做*I/O*处理机，它实际上是一台能力只限于专门控制*I/O*设备进行输入输出的小型计算机，它独立于主机而工作，在计算机系统中，它处于主存和外部设备之间，把主机与外部设备联系起来，构成了如图1-3所示的计算机系统。现代的计算机系统，都配置了*I/O*通道。

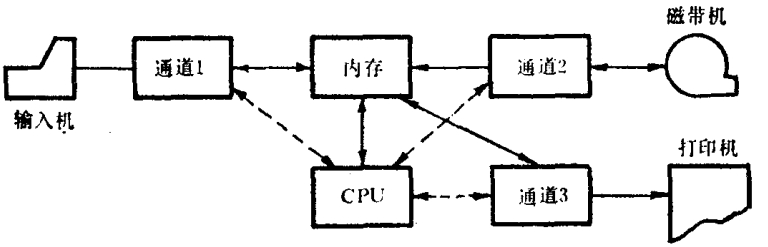

图 1-3　具有I/O通道的计算机系统

*I/O*通道有自己的指令系统和一些寄存器，其指令称为通道指令或通道控制字，它和中央处理机共享同一内存，来存放通道指令和有关信息。当需要*I/O*传送时，将欲传输的信息放入缓冲区后，*CPU*根据*I/O*要求形成由通道指令组成的通道程序(*I/O*程序)，然后通过“启动外设”指令，启动通道执行通道程序，这时，通道便独立于主机而工作，把内存缓冲区中的信息输出到某一*I/O*设备，只有通道把这批信息传送完毕，通道才向*CPU*发出中断，报告这批信息传输结束。有了通道后，*CPU*只在需要时，通知某一通道，要求传输信息，该通道接到通知后，便自行与内存打交道，控制*I/O*设备进行*I/O*操作。在此过程中不再需要中央处理机，实现了中央处理机与外部设备并行工作，外部设备与外部设备之间并行工作。其工作过程示图1-4。

图中表示，当*CPU*执行到时间t_1时，向通道1发出启动*I/O*命

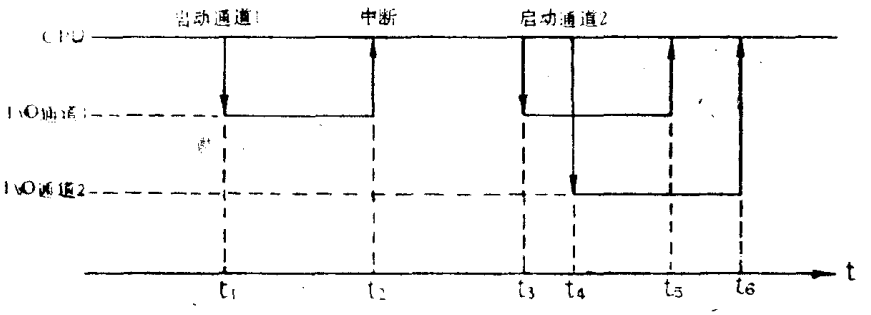

图 1-4　*CPU*与外设并行工作

令，通道1开始工作，在时间t_2，通道1向*CPU*发中断，表示一批信息传输结束，在t_1到t_2这段时间里，*CPU*与通道1所控制的*I/O*设备并行工作。同样，在t_4，*CPU*向通道2发启动*I/O*命令，通道2开始工作，在t_6，通道2所控制的*I/O*设备工作结束，在t_4到t_5这段时间内，通道1和通道2所控制的*I/O*设备并行工作，显然，*CPU*也与它们并行工作。

有了通道后，使*CPU*与外设的关系疏远了，并以存储器为中心的结构取代了以*CPU*为中心的结构，由于主机与*I/O*设备之间可以并行工作，从而减少了中央处理机等待*I/O*传输的停顿时间，解决了慢速*I/O*设备与快速*CPU*之间速度不匹配的矛盾，大大提高了计算机系统的效率。但是，这样的系统中只运行一道作业，即使有了通道，上述的优越性仍体现不出来，计算机系统的资源，还是得不到充分利用的，这是因为，当这道程序由于等待*I/O*设备传输完成，或发生错误时，处理机因无事可作只好空等，如图1-4中的t_1到t_2这段时间里，由于*CPU*等待*I/O*设备传输完成，便无事可做。然而事实上，外部设备是否工作，是由用户程序中的输入输出要求来决定的，而用户的*I/O*要求，在整个运行过程中是不均匀的，以计算为主的程序（*cumputer bound*），要进行大量的计算，而输入/输出要求却很少，该程序运行期间，*I/O*设备几乎闲置；反之，以*I/O*为主（*I/O bound*）的程序，在运行的过程中却频繁地提出输入输出要求，而计算却很少，*CPU*利用率不高。为了充分发挥计算机系统各组成部分的作用，提高其利用率，人们提出了在主机等待*I/O*设备传输时，去运行另一道程序的设想，引入了多道程序设计。

1.1.4 多道程序设计及操作系统的引入

所谓多道程序，就是在内存中同时存放（驻留）着一个以上的彼此无关的程序，当系统中只有一台处理机时，任一时刻，只有一道程序可以被执行，若被执行的那道程序，由于某种原因不能继续

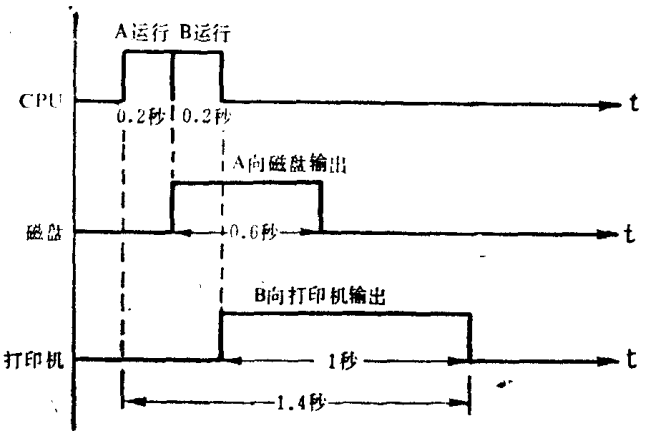

图 1-5　并行操作

运行下去时（如等待 *I/O* 操作完成或出错）系统会记下该程序暂停运行时的“现场”信息，按照一定的原则控制处理机马上转去执行内存中的另一道可运行的程序，当暂停运行的程序等待条件满足时，可恢复其“现场”又继续运行，这样减少了处理机因等待*I/O*完成而停顿的时间，如图1-5所示。与图1-2比较，图1-5中，*A*、*B*作业完成，只需1.4秒的时间，而按图1-2，则需用 2 秒。因此，多道程序是利用驻留在内存中的不同程序在计算时间和 *I/O* 设备执行 *I/O* 操作时间上的重叠，来充分利用*CPU*和各 *I/O* 设备，使其并行工作的。

*I/O*通道和通道中断主机能力的引入，为*CPU*与*I/O*设备并行工作打下了硬件基础，而多道程序设计的引入，才能有效地实现*CPU*与*I/O*设备间、*I/O*设备与*I/O*设备间的并行工作，从而使计算机系统各组成部分的利用率得到尽可能大的提高，因此，多道程序设计的目标，是使多个程序能有效地共享计算机系统资源，使计算机系统的各部件尽可能忙碌地工作。多道程序的方法是比较适应于现代计算机的一种方法，然而，多道程序设计的引入，产生了许多新的矛盾，带来了许多新的问题。操作系统正是解决这些问题而设计的系统软件。这是因为，每道程序在执行时，需要使用各种硬件资源

（*CPU*、内存储器、*I/O* 设备等）和软件资源（各种程序和数据），在多道程序系统中，这些有限的资源，必然为多个程序所“共享”，而资源的共享，不可避免地要引起对资源的“争用”（竞争），为了解决资源的共享和争用问题，人们花费了不少精力，系统也付出了很高的代价。解决资源的共享和争用问题的基本思想，是把系统中的硬件资源虚拟化，即把容量有限的内存改造为使用户感到他在使用一个容量足够大，且为他独占的虚存；把一台共享的物理处理机改造为各用户独占的多台虚拟处理机，把一台慢速的*I/O*设备，改造为多台快速的虚拟 *I/O* 设备等等，其基本方法是如何合理分配各种资源。比如，系统中只有一台处理机，究竟采用什么策略，把处理机分配给内存中哪一个程序最为合理呢？内存是程序得以运行的活动场所，究竟采用什么分配策略，让哪些用户程序能得到内存空间？程序在执行中需要*I/O*设备进行输入输出，系统应采用什么分配策略把设备分配给用户呢？当用户使用完所分配的资源时，又如何释放这些资源等一系列的问题，都应由系统自动解决。操作系统正是为解决这些问题而设计的系统软件。

另外，由于计算机的应用范围不断扩大，必须避免裸机所带来的问题，把计算机从计算机行家手中解脱出来，使之成为广大用户使用方便的工具。

综上所述，随着计算机应用范围的不断扩大，计算机的运算速度不断提高，机内可运行的程序由单道到多道，计算机的功能由弱到强，管理和控制系统的软件由简单到复杂，用户对机器所必须了解的东西由多到少。总之，由于实际应用的需要，促进了计算机软件的发展，形成了操作系统。多道并发程序设计技术，是现代操作系统的重要特征。

1.2 用户与计算机间的接口及操作系统的功能

多道程序的引入，要求设计一整套负责分配各种资源、调度和管理计算机工作流程的程序，以及给用户提供使用方便的计算机接口，因此产生了操作系统。

所谓操作系统，是用来控制和管理计算机系统的软、硬件资源，提高系统资源利用率，组织好计算机工作流程，从而有效使用计算机的各种程序的集合。设置操作系统的目的，是为提高计算机系统的效率，充分发挥各组成部分的作用，让计算机自己管理自己，方便用户。有了操作系统，系统中的全部资源由操作系统统一管理起来，用户不必具体过问各种资源使用状况，例如，不必在编写程序时考虑如何分配内存的问题，也不必为使用 *I/O* 设备而编制复杂的输入输出程序，只要发出简单的命令，整个计算机系统就会在操作系统的控制和指挥下，自动、协调、高效地工作起来，从而大大方便了用户，这时，用户所面对的，不再是一台裸机，而是经操作系统改造后，功能更强的虚处理机，操作系统则成为用户与裸机之间的接口（见图1-6），用户通过操作系统来使用计算机。

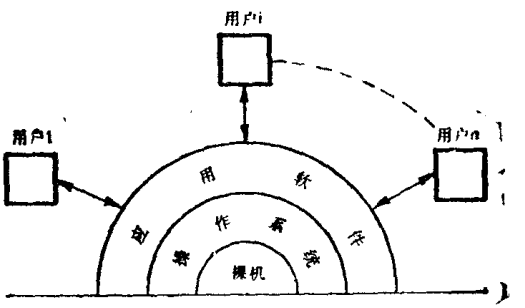

图 1-6 用户与计算机

从资源管理的观点出发，操作系统的基本功能有四个，即：处理机管理、存储管理、外部设备管理、文件管理。前面三个功能对应于计算机系统的三个基本组成部件，文件管理是针对计算机需要处理的数据信息的管理，它向用户提供使用操作系统的手段。

1.3 操作系统的组成成分

操作系统是由一组为提高计算机效率的程序有机地组合起来的。驱动计算机工作的所有程序，都是操作系统的组成成分。在计算机系统中，构成操作系统的软件，可分为两部分，一部分为控制程序，另一部分则为处理程序。如图1-7所示。

处理程序部分：

在操作系统下，为某种目的而设置的程序都定义为处理程序，因此，处理程序包括许多用途不同的程序，如各种语言的编译程

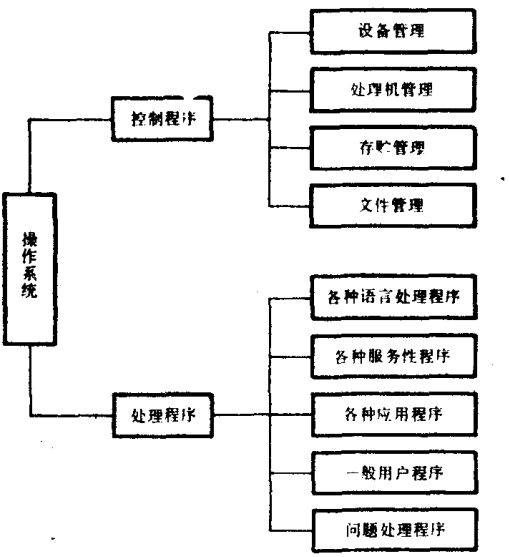

图 1-7　操作系统组成