第一章
操作系统概述

1.1 操作系统的概念

关于什么是操作系统，目前尚无统一的定义。这里只能从操作系统在整个计算机系统中所处的地位以及所起的作用来给出关于操作系统的非形式化描述。

1.1.1 操作系统的地位


操作系统概述

计算机系统是由硬件和软件两部分构成的。软件又分成系统软件和应用软件两类，操作系统（operating system，OS）是一个最基本也是最重要的系统软件。从虚拟机的观点来看，软件是分层次的。系统软件位于低层，应用软件位于高层。当然，对系统软件和应用软件都还可以进一步分层。如果将系统软件进一步分层，可以发现操作系统位于系统软件层次中的最底层，如图 1-1 所示。

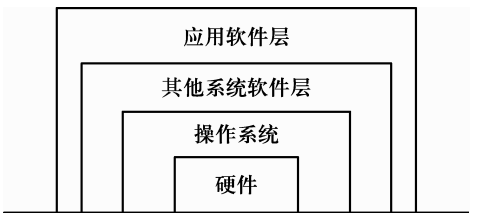

图 1-1　虚拟机层次

据此可以看出，操作系统是与计算机硬件关系最为密切的一个系统软件，是对硬件部件的第一次扩充。

图 1-1 所示的层次关系具有穿透性：高层软件可以调用所有低层的软件，并与硬件直接打交道。每个软件层都在原有层次的基础上增加一层新的界面。例如，应用程序（application program）以目标代码的形式运行时，可以与操作系统和硬件直接打交道（调用操作系统或执行硬件指令），操作系统之上的系统库可以被应用程序调用，系统库中的函数又可以调用操作系统，

如图 1-2 所示。

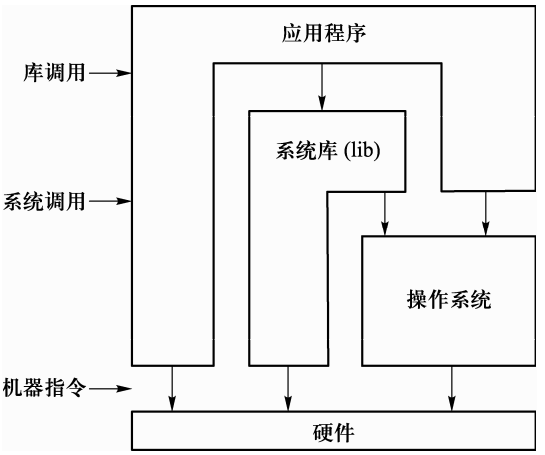

图 1-2 操作系统的地位

1.1.2 操作系统的作用

操作系统有以下两个重要的作用。

1. 管理系统中的各种资源

一个多道计算机系统可以同时为多个用户服务。也就是说，在计算机系统中同时有多个程序在执行。这些程序在执行的过程中会要求使用系统中的各种资源，例如当程序运行时，需要处理器资源，输出结果时需要打印机资源。多个程序的资源需求经常会产生冲突，如程序 1 和程序 2 可能同时要使用打印机进行输出。如果对程序的这些资源需求不加以管理，就会造成混乱甚至损坏设备。也就是说，在系统中需要一个资源仲裁者，由它负责资源在各个程序之间的调度，保证系统中的各种资源得以有效利用。这个资源仲裁者就是操作系统。

2. 为用户提供友好的界面

早期的计算机系统中是没有操作系统的，那时使用计算机需要大量的手动操作，既烦琐又费时。读者可以想象，如果没有操作系统，要运行一个用 C 语言编写的源程序将会多么困难。有了操作系统之后，原来需要由人来做的许多烦琐而又费时的工作可以由操作系统完成，这使得用户能够非常方便地使用计算机系统。例如，要运行一个用 C 语言编写的源程序，用户只需在终端上输入几条命令或者单击几次鼠标即可。可以说，操作系统的产生是计算机发展历程中历史性的一步。

随着硬件成本的不断下降，计算机已经走入家庭和办公自动化领域。计算机的使用者大多不是计算机专业人员，界面的友好性比资源的利用效率更具实际意义。目前商业化操作系统提供的图形用户界面（graphic user interface，GUI）就是在此背景下生成的产物。

1.1.3 操作系统的定义

根据前面关于操作系统的地位和作用的描述，可以给出操作系统的非形式化定义。

定义 1-1 操作系统是位于硬件层之上、所有其他系统软件层之下的一个系统软件，通过它管理系统中的各种软件和硬件资源，使它们能被充分利用，方便用户使用计算机系统。

应当指出，关于操作系统的概念是难以用几句话来概括的，读者需要在学习过程中认真加以体会。

1.2 操作系统的历史

为使读者了解操作系统的形成、完善和发展，下面简略地回顾一下操作系统的历史。

应当指出，由于操作系统是直接建造于硬件层之上的，它的演变必然与计算机系统结构的演变有着密切的联系。可以说，操作系统的发展与硬件系统结构的发展相互促进、相互影响。一方面，为了方便而有效地使用硬件，促进了操作系统的产生；另一方面，为了利于构造操作系统，硬件也经历了不断改进的过程。此外，由于操作系统可以为上层软件及用户提供友好的界面，它的演变必然会反映出上层软件及用户对于操作系统的使用要求。

简言之，操作系统经历了从无到有、从功能单纯到功能完备的演变过程，并且尚处于进一步的发展之中，下面分别加以叙述。

1.2.1 操作系统的产生

计算机操作系统在从无到有的产生过程中经历了以下几个阶段。

1．手动操作阶段（20 世纪 40 年代）

计算机诞生的初期并没有操作系统，人们采用手动操作的方式使用计算机，典型的作业（job）处理步骤如下：首先，将程序和数据通过手动操作记录在穿孔纸带上；然后，将程序穿孔纸带放到光电输入机上，并通过控制台开关启动光电输入机，将程序输入内存；继而再通过控制台开关启动程序由第一条指令开始执行；程序在运行过程中通常需要人工干预，如将穿孔纸带放到光电输入机上，出错时显示错误地址并修改指令等；最后在电传打印机上输出运行结果。显然，这种操作方式有如下两个缺点：用户在作业处理的整个过程中独享系统中的全部资源，手工操作所需的时间很长。

这种操作方式在计算机速度较慢的情况下是可以容忍的，但是当计算机速度大幅度提高之后，就会暴露出严重的缺点。例如，假设一个作业在速度为每秒 1 000 次的机器上运行需要 1 h，手动操作所需要的时间为 4 min，则手动操作时间与程序运行时间之比为 1∶15。若计算机的速度提高到每秒 600 000 次，同样的程序其运行时间只需 6 s，而手动操作时间不变，仍为 4 min，则手动操作时间与程序运行时间之比为 40∶1。也就是说，手动操作时间远远大于程序运行时间。因而，缩短手动操作时间在以晶体管为代表的第二代计算机出现后便成为亟待解决的问题。

其他软件在此阶段所取得的成就主要是汇编语言和汇编系统的出现，这在一定程度上减轻了用户使用计算机的负担。

2．批处理阶段（20 世纪 50 年代）

为了缩短手动操作的时间，人们自然会想到使作业到作业之间的过渡摆脱人为干预，实现自动化，如此便出现了批处理。批处理经历了两个阶段，即联机批处理阶段和脱机批处理阶段。

（1）联机批处理

早期的批处理是联机的。其工作原理如下：操作员将若干个作业合成一批，并将其卡片依次放到读卡机上，监督程序通过内存将这一批作业传送到磁带机上，输入完毕后监督程序开始处理这一批作业。它自动地将第一个作业读入内存，并对该作业的程序进行汇编或编译，然后将所产生的目标程序与所需要的例行子程序连接、装配在一起，继而执行该程序，计算完成之后输出结果。第一个作业处理完毕后立即开始处理第二个作业，如此往复，直到所有作业处理完毕。此时，监督程序将第二批作业由读卡机传送到磁带机上，并按上述步骤进行处理。这样，监督程序不间断地处理各个作业，实现了作业之间转换的自动化，大大地缩短了手动操作的时间。不过，联机批处理也有一个缺点，即作业由读卡机到磁带机的传输需要由处理器完成，由于设备的传输速度远远低于处理器的速度，在此传输过程中处理器仍然会浪费很多时间。联机批处理工作原理如图 1-3 所示。

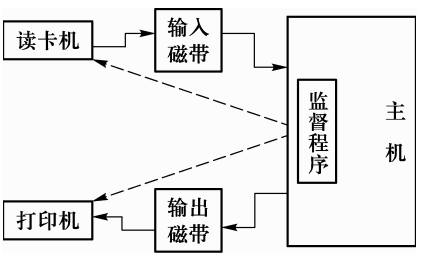

图 1-3 联机批处理

（2）脱机批处理

为了克服联机批处理的缺点，引入了脱机批处理。它的思想是把输入输出操作交给一个功能较为单纯的卫星机去做，把主机从烦琐、耗时的输入输出操作中解脱出来，其基本原理如图 1-4 所示。待处理的作业由卫星机负责经读卡机传送到输入磁带上，一批作业输入完后操作员将磁带由卫星机卸下并安装到主机上，主机由输入磁带读入作业并加以处理，其结果送到输出磁带上，一批结果产生后操作员将输出磁带由主机卸下并安装到卫星机上，最后由卫星机负责将输出磁带上的结果送到打印机上输出。可见，脱机批处理减轻了主机的 I/O 负担，使其能专注于计算任务的处理。其缺点是需要一个专门的卫星机，并且磁带装卸需要人工完成。

批处理系统是操作系统的雏形。在此阶段，其他软件也有了相应的发展，如输入输出标准程序、高级语言编译程序、连接装配程序等。

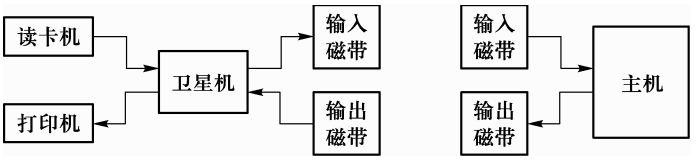

图 1-4 脱机批处理

3．执行系统阶段（20 世纪 60 年代初期）

批处理较手动操作来说前进了一大步，但是它仍然存在一些缺点，如需要额外的卫星机、磁带机的装卸需要手动操作等。

在 20 世纪 60 年代初期，硬件在两个方面取得了重要的进展。一是引入通道，二是出现通道中断主机功能，这是操作系统发展史上的重要事件，它推动操作系统进入执行系统阶段。

通道又称 I/O 处理器，它具有自己的指令系统和运算控制部件，与处理器共享内存资源。通道可以受处理器的委托执行通道程序以完成输入输出操作，通道的输入输出操作可以同处理器的计算工作完全并行执行，并在输入输出操作完成时向处理器发出中断请求。这样，作业由读卡机到磁带机的传输以及结果由磁带机到打印机的传输均可由通道完成，这既非联机方式，也非脱机方式，称为“假脱机”（simultaneous peripheral operations on line，SPOOL）或“伪脱机”。通道取代了卫星机，也免去了手动装卸磁带的麻烦。

执行系统阶段是操作系统的初级阶段，它为操作系统的最终形成奠定了基础。

1.2.2 操作系统的完善

操作系统由形成到完善经历了以下几个主要发展过程。

1．多道批处理系统（20 世纪 60 年代初期）

执行系统出现不久，人们就发现在内存中同时存放多道作业是有利的。当一道作业因为等待 I/O 传输完成而暂时不能运行时，系统可以将处理器资源分配给另一个可以运行的程序，如此便产生了多道批处理系统。

多道批处理的出现是操作系统发展史上一个革命性的变革，它将多道程序设计的概念引入操作系统中。后面将会讲到，多道程序设计与传统的单道程序设计相比具有本质的差别，它的引入在理论及实践等方面给操作系统带来了许多新的研究课题。

2．分时系统（20 世纪 60 年代初期和中期）

手动操作是一种联机操作方式，其效率很低。批处理系统替代了手动操作，是一种脱机操作方式。执行系统及多道批处理系统是批处理系统的进一步发展，属于更高级的脱机处理方式。但是，多道批处理系统出现不久，人们便发现仍有联机操作的必要，这是由程序员首先提出的。对于脱机操作来说，程序员无法了解作业的执行情况，无法对其进行动态控制。如果作业在处理过程中出现错误，程序员不能对其及时进行修改，必须等待批处理结果输出后才能从输出报告中得知错误所在，并对其进行修改，然后再次提交批作业，如此可能需要重复多次，使得作

业的处理周期比较长。也就是说，脱机方式非常不利于程序的动态调试。

为达到联机操作的目标，出现了分时系统。分时系统由一台主机和若干台与其相连的终端构成，用户可以在终端上输入和运行程序，系统采用对话的方式为各台终端上的用户服务，便于程序的动态修改和调试，缩短了程序的处理周期。由于多个终端用户可以同时使用同一个系统，因而分时系统也是以多道程序设计为基础的。

多道批处理系统与分时系统各有千秋，前者适用于处理大型科学计算任务，后者适用于处理交互式任务。它们是现代操作系统的两大主要类别。多道批处理系统和分时系统的出现标志着操作系统已经进入完善阶段。

3．实时处理系统（20 世纪 60 年代中期）

在 20 世纪 60 年代中期，集成电路取代了分立元件，计算机进入第三代。由于计算机性能的提高，计算机的应用范围迅速扩大，从传统的科学计算扩展到商业数据处理，进而深入各行各业，例如工业自动化控制、医疗诊断、航班订票等，这样就出现了实时操作系统。

多道批处理系统、分时系统和实时处理系统是传统操作系统的三大类别，它们为通用操作系统的最终形成做好了必要的准备。

4．通用操作系统（20 世纪 60 年代后期）

为了进一步提高计算机系统的适应能力和使用效率，人们将多道批处理、分时和实时等功能结合在一起，构造出多功能的通用操作系统。通用操作系统可以同时处理实时任务、接收终端请求、运行成批作业。当然，通用操作系统更加庞大、更加复杂，造价也更高。

1.2.3 操作系统的发展

20 世纪六七十年代至今，人们已经成功地设计出许多优秀的实用操作系统，例如 Atlas（英国曼彻斯特大学）、XDS-940（美国加州大学伯克利分校）、THE（荷兰 E.W. Dijkstra）、RC-4000（丹麦 P. B. Hansen）、CTSS（美国麻省理工学院）、Multics（美国麻省理工学院）、OS/360（美国 IBM 公司）、VMS（美国 DEC 公司）、GCOS-8（美国 Honeywell 公司）、B7000 MCP（德国 Broughs 公司）、OS/2（美国 IBM 公司）、UNIX（美国贝尔实验室）、Windows（美国微软公司）、Linux（自由软件）等。

50 多年来，操作系统在多方面取得了更大的发展，主要表现在以下几个方面。

① 硬件体系结构由集中向分散发展，出现了计算机网络。计算机网络是计算机技术与现代通信技术相结合的产物，它突破了空间上的限制，将地理上分散、功能各异的计算机连接在一起，形成一个相对完整、功能更加强大的计算机系统。计算机网络在商业、文化、教育、通信、管理、军事等各个领域为计算机的应用开辟了更加广阔的前景，成为现代信息化社会最重要的工具，也为操作系统的研究提出了新的课题，即如何有效地管理网络资源，并实现分布式环境中的并发控制。为此出现了网络操作系统和分布式操作系统，由此而带来的计算机安全问题引起了操作系统研究者的普遍关注。

② 随着微处理器技术的迅猛发展，家庭和商用微型计算机得到了普及。这些微型计算机

具有性能较好、价格低廉等特点，并且提供了友好的操作界面。在这种微型计算机上，单用户多任务的操作系统占主导地位。

③ 在科学和军事领域，大型计算任务要求极强的计算和处理能力，在单一处理器的计算能力不能满足处理要求的条件下，多处理器并行成为必然选择。多处理器并行使并发控制问题变得更加复杂，由此产生了并行操作系统。随着处理器芯片的降价，服务器甚至微机主板上都配有多个处理器插槽，多数流行操作系统也提供了相应的支持功能。

④ 传统的操作系统是以计算机为中心的。随着处理器芯片和各种存储介质在控制领域的广泛应用，嵌入式和智能卡操作系统应运而生。在这些领域中，“计算”是为某种具体应用服务的，处于附属地位。应用的多样化要求操作系统具有专用特性，然而为每个具体应用开发一个操作系统的代价过高，因而人们尝试从不同的应用中抽取具有共性的内容，由此构成很小的操作系统核心。继而产生了微内核操作系统体系结构。

⑤ 伴随后摩尔定律时代的到来，提高单处理器速度已近极限，多核技术应运而生。如何管理多个核心以提高系统性能成为新一代操作系统急需解决的问题。多核系统主要带来两个问题：一是并发控制，单处理器和单核系统的并发控制是在指令级别上的，而多处理器和多核的并发控制是在指令周期级别上的；二是调度，如何合理安排多线程在多处理器或多核上运行使其达到更高的性能。

⑥ 随着集群系统和云计算的出现，产生了集群操作系统和云计算操作系统，它们都属于并行操作系统的范畴。

1.3 操作系统的特性


操作系统特性

操作系统也是一个程序，它具有 4 个非常重要的特性，即程序的并发性（concurrency）、资源的共享性（sharing）、异步性（asynchronous）和虚拟性（virtuality），把握操作系统的这 4 个特性对于深刻理解操作系统会有很大帮助。

1.3.1 并发性

所谓程序并发，是指在计算机系统中同时存在多个程序。从宏观上看来，这些程序是同时向前推进的。程序的并发性具体体现在以下 3 个方面：用户程序与用户程序之间并发执行，用户程序与操作系统之间并发执行，操作系统与操作系统之间并发执行。

这里需要区别两个相关但不完全相同的概念，即程序并发与程序并行。程序并行要求微观上的同时，即在绝对的同一时刻有多个程序同时向前推进；而程序并发并不要求微观上的同时，只需从宏观上看多个程序都在向前推进。显然，要实现程序并行必须要有多个处理器，但是在单处理器环境中可以实现程序并发，此时这些并发执行的程序是按照某种次序交替地获得处理器并运行的。由于处理器的速度很快，因而从宏观上看，这些程序都在向前推进，仿佛每个程序都拥有一个属于自己的处理器，即所谓的虚处理器。

在算法研究的范畴通常使用“并行”这个术语，而在单处理器操作系统中通常使用“并发”这个术语，尽管处理器与设备之间、设备与设备之间可以并行工作。

1.3.2 共享性

所谓资源共享是指操作系统与多个用户程序共用系统中的各种资源，这种共享是在操作系统的控制下实现的。当然为实现这种控制，操作系统也需要消耗资源，这些资源包括处理器、主存储器、外存储器（简称外存）、设备、信息等。对于一个给定的计算机系统来说，它的资源配置情况是相对固定的，而程序对于资源的需求则是变化的，且通常是不可预知的。操作系统要掌握系统中当前资源的使用情况，并据此决定各个程序进入系统的次序以及使用资源的次序。

1.3.3 异步性

在操作系统之上，宏观上同时运行的程序有多个，这些程序（连同操作系统程序）是交替执行的。交替的切换点是中断，中断使用户程序切换到操作系统程序，嵌套中断使一段操作系统程序切换到另一段操作系统程序，而中断的发生时刻却是不确定的，因而操作系统的运行轨迹是异步的、不可预知的。应当指出，异步性并非一个独立的特性，它实际上是由并发派生出来的一个特性，即并发性必然导致随机性和异步性，因而并发性与异步性不是两个正交的特性。

1.3.4 虚拟性

所谓虚拟（virtual），是利用某种技术把一个物理实体变为若干个逻辑实体。物理实体是实际存在的，而逻辑实体则是虚化的。虚拟技术在操作系统中被广泛采用，例如，为在单处理器系统中同时（宏观）运行多个程序，操作系统把一个实的 CPU 改造成多个虚的 CPU，每个用户（程序）仿佛拥有一个属于自己的 CPU；为打破内存价格高、容量小的限制，利用外存空间实现虚拟存储，使用户感觉存储容量相当于外存储器、速度接近内存；为解决设备数量少、速度慢的问题，利用快速存储型设备构造数量多、速度快的虚拟设备。

实际上，操作系统就是一个大的虚拟机，这个大的虚拟机可以通过一层层的虚拟扩充来实现，后面在操作系统设计中还会进一步加以讨论。

1.4 操作系统的分类

按照操作系统的功能可将其分为以下几类：多道批处理操作系统、分时操作系统、实时操作系统、通用操作系统、单用户操作系统、网络操作系统、分布式操作系统、多处理器操作系统、集群操作系统、云计算操作系统嵌入式操作系统、多媒体操作系统、智能卡操作系统等。下面分别作简略介绍。

1.4.1 多道批处理操作系统

多道批处理操作系统（batch processing operating system）是以脱机操作为标志的操作系统，

特别适合于处理运行时间比较长的程序，其工作原理如图 1-5 所示。

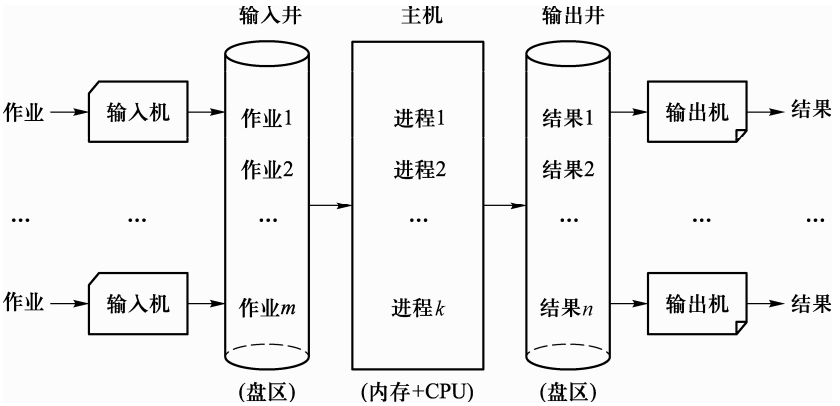

图 1-5 多道批处理操作系统工作原理

在使用这种系统时，用户无法对其程序的运行状况施行交互性控制。当他将一个计算任务交给系统处理时，必须将其控制意图“告诉”操作系统，如第一步做什么，第二步做什么，出错时如何处理等。为此，他需要用操作系统所提供的作业控制语言书写一份说明书，该说明书称为作业说明书，并将其与程序和数据一并交给系统。操作系统按照作业说明书所规定的步骤完成相应的计算任务。用户程序、数据以及作业说明书合称为作业。

批作业的处理步骤如下：用户将作业（程序、数据、说明书）交给机房工作人员，操作员在适当的时刻将其放到某台输入机上并启动其工作，通道负责将作业传输到磁盘输入井中，以后在适当的时刻经通道传输进入内存处理。此时作业以“进程”为单位在内存中运行，运行结束后，其结果经通道传输进入磁盘输出井中。最后，再由通道负责将结果在输出机上以用户可见的形式显示出来。

输入井和输出井分别为磁盘或磁鼓上的两个区域，输入井用于保存已经输入但尚未处理的作业；输出井用于保存处理完毕但尚未输出的结果。设置输入井和输出井的目的主要有两个：协调输入输出设备速度与处理器速度之间的差异；为作业调度提供有利条件，如果没有输入井，系统只能按照自然次序处理作业，设置输入井后，系统可以根据调度的需要在输入井中选择进入内存的作业，使得内存中运行的作业搭配合理。

多道批处理操作系统具有两个特性。

① 多道。内存中同时存在多个正在处理的作业，而且外存储器输入井中还有多个尚待处理的作业。

② 成批。作业逐批地进入系统，逐批地处理，逐批地离开系统。作业与作业之间的过渡由操作系统控制，无须用户干预。

1.4.2 分时操作系统

分时操作系统（time-sharing operating system）是以联机操作为标志的操作系统，特别适合

于程序的动态调试和修改。

在一个分时系统中，一个主机同多个交互终端相连，这些终端既可能是本地的，也可能是远程的。每个终端上可以有一个用户，系统以对话的方式与终端用户交互，如图 1-6 所示。

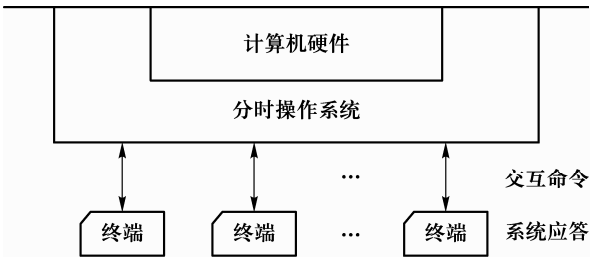

图 1-6 分时操作系统

分时操作系统为终端用户提供一组交互终端命令，它是用户与操作系统之间交互的界面。用户进入系统后，可以在终端上输入终端命令，该命令被操作系统接收，后者执行一段系统程序，完成用户交付的任务，然后给出一个应答，用户根据应答确定下一个将要输入的命令。如此往复，直至用户完成其计算任务后退出系统。

这类系统是采取分时的方法为多个终端用户提供服务的，它将时间划分为若干个片段，称为时间片，并以时间片为单位轮流地为各个交互终端用户服务。由于时间片通常很短，如十几毫秒或几十毫秒，系统为所有用户服务一次仅需较短的时间。例如，对于一个拥有 50 个终端的系统来说，假设时间片的长度为 40 ms，一个终端每隔 2 s 左右便能得到一次系统响应。

分时操作系统具有以下 3 个重要的特性。

① 多路性。又称多路调制性，即一个主机可以同时与多个终端相连。根据硬件配置情况，同一主机可以与几个、十几个以至数十个，甚至上百个终端连接在一起。

② 交互性。又称交往性，即系统以对话的方式为各个终端用户服务。用户在终端上可以方便地录入、调试、修改、运行其程序。

③ 独占性。由于计算机的运行速度很快，相比之下手动操作的速度较慢，因而每个用户感觉仿佛独占整个计算机系统，而不知道其他用户的存在，即每个终端用户实际上都拥有一台完全属于自己的虚拟机。

1.4.3 实时操作系统

所谓实时，是指系统能够对外部请求做出及时的响应。实时操作系统（real-time operating system）按其应用范围可以分为实时控制和实时信息处理两大类别。

1. 实时控制

实时控制包括工业控制、宇航控制、医疗控制、铁路运输控制等。这类系统都有一个被控制对象，被控制对象所产生的信号通过特殊的外围设备（又称外部设备，简称外设）传送给计算机系统，计算机系统接收来自被控制对象的请求信号后对其进行分析，并做出处理决策，然

后将控制信号通过特殊的外围设备传送给被控制对象，如图 1-7 所示。

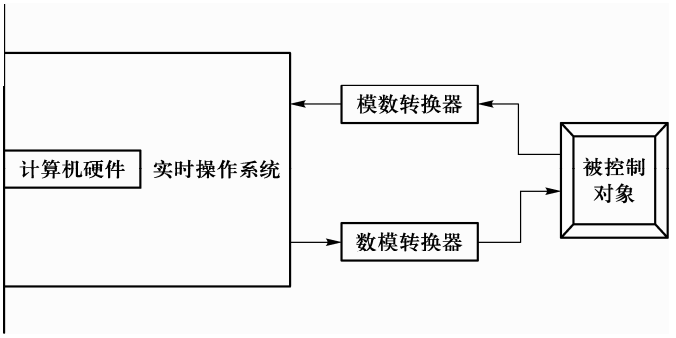

图 1-7 实时控制

由于被控制对象所产生的信号以及所接收的信号多为模拟量，信号由被控制对象传送到实时系统需要经过模数转换，信号由实时操作系统传送到被控制对象需要经过数模转换。

2．实时信息处理

实时信息处理包括联机情报检索、图书管理、航班订票等。这类系统的一般原理与分时系统相似，不过相连终端通常是远程的，可能分散于一所学校、一座城市、整个国家的不同地区，甚至跨越国界。

容易理解，实时操作系统应当具有两个基本特性。

① 及时性。即能够对外部请求做出及时的响应和处理。实时操作系统要求响应速度，但这个响应速度是一个相对的量。例如，对于实时控制来说，不同的被控制对象要求的响应速度是不同的，有的可能在毫秒级，有的可能在秒级，实时操作系统应当能够在被控制对象可以容忍的时间范围内对外部请求做出响应和处理。如果一个实时操作系统可以同时处理多种实时任务，应当保证所有这些任务在规定的截止期内处理完毕。

② 可靠性。与其他类型的系统相比，实时操作系统更加注重其稳定性和可靠性。例如对于航天控制系统来说，实时控制系统的故障可能带来的后果是无法估量的。

1.4.4 通用操作系统

同时具有分时、实时和批处理功能的操作系统称为通用操作系统（general purpose operating system）。显然，通用操作系统规模更加庞大，构造更加复杂，功能也更加强大。构造通用操作系统的目的是为用户提供多模式的服务，同时进一步提高系统资源的利用效率。

在通用操作系统中，可能同时存在 3 类任务，即实时任务、分时任务、批处理任务。这 3 类任务通常按照其急迫程度加以分级：实时任务级别最高，分时任务级别次之，批处理任务级别最低。当有实时请求时，系统优先处理；当没有实时任务时，系统为分时用户服务，仅当既无实时任务也无分时任务时，系统才执行批处理任务。

在实际应用中，同时具有实时、分时、批处理 3 种功能的操作系统并不常见。通常将实时与批处理结合起来，或者将分时与批处理结合起来，构成所谓的前后台系统。在实时与批处理相结合的系统中，实时任务为前台，批处理任务为后台；在分时与批处理相结合的系统（如 GCOS-8）中，分时任务为前台，批处理任务为后台。前台任务优先于后台任务。

1.4.5 单用户操作系统

单用户操作系统（single-user operating system）是为个人计算机所配置的操作系统。这类操作系统最主要的特点是单用户，即系统在同一段时间内仅为一个用户提供服务。早期的单用户操作系统（如 MS-DOS）以单任务为主要特征，由于一个用户（程序）独占整个计算机系统，操作系统资源管理的任务变得不重要，为用户提供友好的工作环境成为这类操作系统更主要的目标。现代的单用户操作系统，如 Windows，已经广泛支持多道程序并发和资源共享。由于单用户操作系统应用广泛，使用者大多不是计算机专业人员，所以一般更加注重用户的友好性和操作的方便性。

常见的单用户操作系统有 MS-DOS、CP/M、Windows 等。单用户操作系统的设计及实现可以采用多道批处理操作系统中所采用的技术，如多进程和多线程、虚拟存储管理方式、层次结构文件系统等。

1.4.6 网络操作系统

用于实现网络通信和网络资源管理的操作系统称为网络操作系统（network operating system，NOS）。网络中的主机以及相连的外围设备称为 HOST，它们可能属于不同类型。各个 HOST 上配置有网络操作系统，它们的界面形式可能有所不同，各个结点具有自治性。图 1-8 给出一个仅由 3 台 HOST 所构成的简单网络操作系统，其中虚线为通信线路，它们将网络中的 HOST 连接起来。

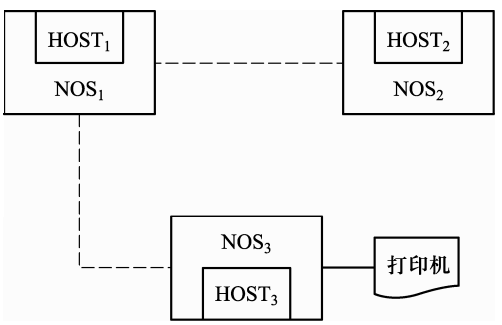

图 1-8 网络操作系统

建造网络操作系统主要有两个目的：相互通信，处于不同 HOST 上的用户之间可以通过网络任意传递信息；共享资源，网络操作系统为资源共享提供了更大的潜力，一个用户不仅可以

使用本地 HOST 上的资源，而且可以使用其他 HOST 上的资源，例如，处于 $HOST_2$ 上的用户可以使用 $HOST_3$ 上所配置的打印机，处于 $HOST_3$ 上的用户可以使用 $HOST_1$ 上存储的数据。

1.4.7 分布式操作系统

分布式操作系统（distributed operating system，DOS）分为两类：一类建立在多机系统的基础之上，称为紧耦合（closely coupled）分布式系统；另一类建立在计算机网络的基础之上，称为松散耦合（loosely coupled）分布式系统。图 1-9 给出了一个松散耦合分布式操作系统的结构。

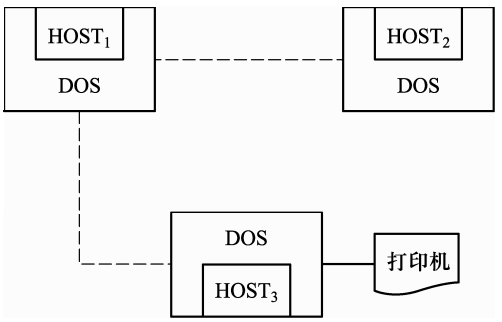

图 1-9 分布式操作系统

分布式操作系统是网络操作系统的更高级形式，它保持网络操作系统所拥有的全部功能，同时具备如下特征。

① 统一的操作系统。对于网络操作系统来说，不同 HOST 上所配置的 NOS 其界面形式可能是不同的；而在分布式操作系统中，所有 HOST 的 DOS 统一，其界面相同。

② 资源的进一步共享。在网络操作系统中，由于各 HOST 上的操作系统不统一，因而一个计算任务不能由一台 HOST 任意迁移到另一台 HOST 上运行；而在分布式操作系统中，所有 HOST 的操作系统界面一致，作业可以由一台 HOST 任意迁移到另一台 HOST 上处理，即可实现处理器资源的共享，从而达到整个系统的负载平衡。

③ 可靠性。构成分布式系统的不同 HOST 处于等同的地位，即没有主从关系，任何一个 HOST 的失效都不会影响整个系统。

④ 透明性。网络用户能够感觉到本地 HOST 与非本地 HOST 在地理位置上的差异；而在分布式系统中，所有 HOST 构成一个完整的、功能更加强大的计算机系统，操作系统掩盖了不同 HOST 在地理位置上的差异。

许多现代操作系统都提供分布处理功能，如 Solaris MC。

1.4.8 多处理器操作系统

具有公共内存和公共时钟的多 CPU 系统称为多处理器操作系统（multiprocessor operating system），也称为紧耦合系统。建立在多处理器上的操作系统也称为并行操作系统，如图 1-10

所示。多 CPU 系统中的多个 CPU 若型号和地位相同，没有主从关系，则称之为对称多处理（symmetric multiprocessing，SMP）的。对称多处理是多处理器操作系统的主要形式。

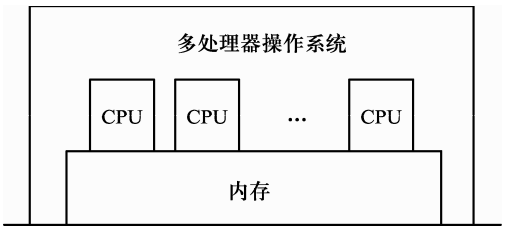

图 1-10 多处理器操作系统

从资源管理的角度来看，处理器和打印机同属于系统资源，但是打印机属于被动型资源，而 CPU 则属于主动型资源。由一台打印机增加为两台或多台打印机，管理的复杂程度并没有明显不同，而由一个 CPU 增加为两个或多个 CPU 其复杂程度却有质的变化，其原因是不仅要进行多个处理器的资源管理，而更主要的是要进行多处理器的并发控制，因而多处理器环境需要专门的操作系统。由于目前多处理器已经被服务器广为采用，现代操作系统如 UNIX、Linux、Windows 都增加了多处理器管理的功能。有关并行操作系统的理论可以单独开设一门课程。

1.4.9 集群操作系统

集群操作系统（cluster operating system）建立在局域网络基础上，是指一种由多台计算机通过软件互相连接组成的并行或分布式系统，可以作为单独、统一的计算资源来使用。

集群的关键技术分为网络层、结点机及操作系统层、集群系统管理层、应用层 4 个层次。这 4 个层次技术有机结合，所有的相关技术虽然解决的问题不同，但都有其不可或缺的重要性。集群系统管理层是集群系统所特有的功能与技术的体现。在未来按需计算的时代，每个集群都应成为业务网格中的一个结点，所以自治性（自我保护、自我配置、自我优化、自我治疗）也将成为集群的一个重要特征。自治性的实现，各种应用程序的开发与运行，大部分直接依赖于集群的系统管理层，并且，系统管理层的完善程度也决定了集群系统的易用性、稳定性、可扩展性等诸多关键参数。正是集群管理系统将多台机器组织起来，使之可以被称为“集群”。

分布式是指将不同的业务分布在不同的地方。而集群指的是将几台服务器集中在一起，实现同一业务。分布式结构中的每一个结点都可以做集群，而集群并不一定就是分布式的。

1.4.10 云计算操作系统

云的思想是把互联网中计算能力比较弱的各个结点统一管理起来，构成一个功能强大的计算机系统。云计算操作系统（cloud computing operating system），又称云操作系统、云计算中心操作系统，是以云计算、云存储技术作为支撑的操作系统，是云计算后台数据中心的整体管理运营系统（也有人认为云计算系统包括云终端操作系统，例如现在流行的各类手机操作系统，

这与单机操作系统区别不大），它是指构架于服务器、存储、网络等基础硬件资源和单机操作系统、中间件、数据库等基础软件之上的，管理海量的基础硬件、软件资源的云平台综合管理系统。

1.4.11 嵌入式操作系统

在机器人、个人数字助理（personal digital assistant，PDA）、车载系统、家用电器、手机等通信设备上，都需要一个支持多道程序设计的环境，提供这种环境的操作系统称为嵌入式操作系统（embedded operating system）。嵌入式操作系统大多用于控制，因而具有实时特性。嵌入式操作系统与一般操作系统相比在以下几方面具有比较明显的差别。

1．可裁减性

因为嵌入式操作系统的硬件配置和应用需求的差别很大，要求嵌入式操作系统必须具备比较好的适应性，即可裁减。在一些配置较高、功能要求较多的环境中，能够通过加载较多的模块来满足这种需求；而在配置较低、功能单一的环境中，系统必须能够通过裁减方式把一些不需要的模块裁减掉。

2．可移植性

在嵌入式开发中，存在多种多样的 CPU 和低层硬件环境，仅流行的 CPU 就会达到十几款，在设计时必须充分考虑，通过一种可移植方案来实现不同硬件平台上的移植。例如，可以把硬件相关部分单独剥离出来，在一个相对独立的模块或源文件中实现，或者增加一个硬件抽象层来实现不同硬件的底层屏蔽。

3．可扩展性

这是指可以很容易地在嵌入式操作系统上扩展新的功能。例如，随着 Internet 的快速发展，可以根据需要，在对嵌入式系统不做大量改动的情况下，增加 TCP/IP 协议功能和 HTTP 协议解析功能。这样要求在进行嵌入式系统设计时，充分考虑功能之间的独立性，并为将来的扩展预留接口。

因为存在上述差别，嵌入式操作系统一般采用微内核（micro kernel）结构。所谓微内核就是非常小巧的操作系统核心，其中只包含绝对必要的操作系统功能，其他功能（如与应用有关的设备驱动程序）则作为应用服务程序置于核心之上，并在目态下运行。当然也有采用单核（monolithic）结构的嵌入式操作系统，这种结构虽运行速度快，但是适应性不及微内核结构。

尽管目前微内核尚无统一的规范，但一般认为内核应当包括如下功能：处理器调度，基本内存管理，通信机制，电源管理。而虚拟存储管理、文件系统、设备驱动程序等则处于核心之外，以目态形式运行。随着嵌入式系统的发展和成熟，有理由相信在不远的未来将会形成相应的工业标准。

微内核结构的明显优点是可靠性高、可移植性好。然而它也有不可忽视的缺点，即系统效率低。应用程序关于文件和设备的操作一般需要经过操作系统转到另一个应用程序，然后再返回到原来的应用程序，其中涉及两次进程切换。

嵌入式操作系统具有微小、实时、专业、可靠、易裁减等优点。具有代表性的嵌入式操作系统有 WinCE（微软公司的 Vinus 计划）、PalmOS、μC Linux、VxWorks、国内的 Hopen（女娲计划）等。

1.4.12 多媒体操作系统

多媒体操作系统是指除具有一般操作系统的功能外，还具有多媒体低层扩充模块，支持高层多媒体信息的采集、编辑、播放和传输等处理功能的系统。

多媒体操作系统大致可分为 3 类：具有编辑和播放双重功能的开发系统，以具备交互播放功能为主的教育/培训系统，用于家庭娱乐和学习的家用多媒体系统。

1.4.13 智能卡操作系统

最小的操作系统当数智能卡操作系统（smart card operating system），这种如信用卡大小的设备上包含一个 CPU 芯片，但是对 CPU 的计算能力和存储容量都有严格的限制。最简单的系统只能执行单一功能，如电子支付，功能较强的系统可以完成多个功能，通常都属于个人私用系统。

有些智能卡是面向 Java 的，即在其智能卡的 ROM 中保存一个 Java 虚拟机（Java virtual machine，JVM）的解释程序，可以将 Java 小应用程序下载到智能卡上并由 JVM 解释执行。有些智能卡可以同时处理多个 Java 小应用程序，多任务调度成为必需的功能。资源管理和保护也是多个小应用程序并发执行的必要条件，这些功能必须由智能卡上的操作系统来提供。

1.5 操作系统的硬件环境


操作系统运行环境

本节将介绍为构造一个高效、可靠的操作系统，硬件需要提供哪些支持。这些内容在有关硬件的课程中可能已讲过，但我们是从操作系统的角度来强调这些知识，这对理解后续知识是很必要的。

1.5.1 定时装置

为实现系统的管理和维护，硬件必须提供定时装置，也就是实时时钟。硬件时钟通常有两种，即绝对时钟和间隔时钟。

1．绝对时钟

绝对时钟类似于电子表，其时间表示形式为年：月：日：时：分：秒。绝对时钟的值保存于硬件寄存器中，开机时可由电源供电计时，关机时可由机内电池供电计时，其值可由程序设定和修改，但是一般通过特权指令完成。当然，程序也可以读取绝对时钟的值。

绝对时钟是必要的，操作系统需要根据绝对时钟的值记录作业进入系统和处理的时间、文件的修改和存取时间、资源占用时间、日志记录时间等。

2. 间隔时钟

间隔时钟也称为闹钟，它每隔固定的时间，如 10 ms，发生一次时钟中断。时钟中断发生后，操作系统获得系统的控制权，以便运行系统管理和实现程序并发。后面将会看到，中断是系统并发的必要条件，即只有通过中断才能实现多道程序设计。尽管除时钟外还有其他可能引起中断的事件，但是那些事件是否发生、何时发生都不确定，而只有时钟中断是最“忠实可靠”的，可见间隔时钟是现代操作系统的基础。

利用间隔时钟还可以实现逻辑时钟，例如，要实现一个 50 ms 的逻辑时钟，可以设置一个初值为 5 的变量，每次时钟中断将变量值减 1，当其值减至 0 时即达到 50 ms。

1.5.2 堆与栈

在计算机学科领域中，“堆”和“栈”是两种不同的数据结构，用途截然不同，尽管用户进程的“堆”和“栈”在物理上通常是相邻的。每个运行程序都有一个堆和两个栈（一个用户栈，一个系统栈）。

堆属于用户空间，用于保存程序中的动态变量，例如树的结点，堆空间由操作系统分给运行程序，由于不同程序运行时对动态变量的使用情况不尽相同，因而堆空间大小需求不定。为了既节省空间又能满足所有程序对堆空间的需求，操作系统一般为运行程序分配一个基本大小的堆，如 8 KB，如果程序运行时对堆空间的需求没有超过 8 KB，则不需要操作系统进一步的帮助，对堆空间的使用由应用软件管理；但如果用户程序运行时对堆的需求超过了 8 KB，则会发生中断，操作系统为其扩充堆空间，如再分配 8 KB，如此继续下去。

用户栈属于用户空间，用于保存用户函数调用时的返回点、参数、局部变量、返回值。除此之外用户堆还有一个用途，即传送调用操作系统时传给操作系统的参数。用户程序调用操作系统时，有两个载体可以用来传递参数：寄存器和用户栈，对于比较小的数据，如一个字符、一个整数、一个浮点数，可以通过寄存器传递；但比较长的参数，如文件名（字符串），寄存器通常是放不下的，只能通过用户栈传递。对每个系统调用，操作系统都规定了参数和返回值的存放位置，用户程序必须遵循相应的规定。

系统栈也称核心栈，系统栈在逻辑上属于操作系统空间，主要有两个用途：一是保存操作系统子程序间相互调用的返回点、参数、局部变量、返回值，当然长度较小的参数和返回值也可以通过通用寄存器传递，这个用途与用户栈相似；二是中断响应时保存中断现场，包括 PSW 和 PC，以及中断处理过程中用到的寄存器值。对于嵌套中断，被中断程序的现场信息依次压入系统栈，中断返回时逆序弹出。

程序切换的同时伴随着堆、用户栈和系统栈的切换，但硬件的栈指针是多个进程共享的。

1.5.3 寄存器

硬件系统提供一套寄存器，由运行进程使用。程序切换时，一般需要把寄存器的当前值保存起来，再次运行前再恢复。这些寄存器简介如下。

① 程序状态字（program status word，PSW）：如图 1-11 所示，IBM 360/370 程序状态字由 16 B 组成，表示当前程序的运行环境，其中每个“X”表示 1 B（4 b）。其中，前 8 位为系统屏蔽位，第 0～6 位分别对应 7 个通道通道，第 7 位对应外中断；第 12～15 位为 CMWP，第 13 位为开关中断位 M，第 15 位为系统状态位 P；第 16～31 位为中断码，用于保存中断字（详细中断信息）；第 36～38 位为程序屏蔽位，分别对应定点溢出、十进溢出、阶下溢；第 39 位备用。可以看出，有些中断是不可屏蔽的，如时钟、地址越界、缺页、非法指令。

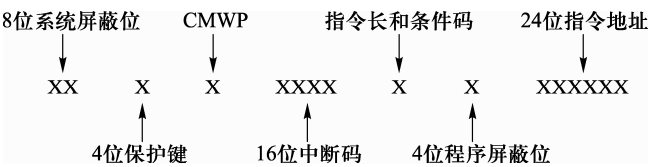

图 1-11 程序状态字组成

② 指令计数器（PC），记载运行程序下一条指令的地址。

③ 栈指针 SP，管态与目态各一个，分别保存系统栈和用户栈的栈顶位置。

④ 通用寄存器（regs），若干个，用于存数和计算，还可用来保存系统调用时传给操作系统的参数，以及由操作系统传给用户的返回值。

⑤ 浮点寄存器（fregs），若干个，用于存数和计算，也可用来保存系统调用时传给操作系统的参数，以及由操作系统传给用户的返回值。

⑥ 地址映射寄存器，一般有一对，分别记录内存区域的起始地址和长度，分别称为基址寄存器（base）和限长寄存器（limit）。

1.5.4 特权指令与非特权指令

现代计算机的指令系统由特权指令集和非特权指令集两个部分组成，它们的使用与系统状态有关。

1．特权指令

只能在管态下才能执行的指令称为特权指令（privileged instruction），如开关中断、修改地址映射寄存器、置程序状态字、停机等。这些指令的执行不仅影响运行程序本身，也会影响其他程序，甚至整个系统。这些指令一般只有操作系统才能执行，而一般用户程序不可执行。

2．非特权指令

在管态和目态下均可执行的指令称为非特权指令（non-privileged instruction），这些指令的

执行只与运行程序本身有关，不会影响其他程序和操作系统，例如数据传送指令、算术运算指令等。

1.5.5 处理器状态及状态转换

1. 处理器状态

为构造一个可靠的系统，硬件至少要区分两种状态：管态和目态，它由一位触发器标识，通常属于程序状态字的一部分，即由程序状态字中的一位标识。

（1）管态

管态（supervisor mode）也称为系统态（system mode）、核心态（kernel mode），是操作系统运行时所处的状态。计算机处于管态时可以执行硬件所提供的全部指令，包括特权指令和非特权指令。由于利用特权指令可以修改程序状态字，而机器状态是程序状态字的一部分，因而在管态下可以改变机器状态，从而可以由管态转换为目态。

（2）目态

目态（object mode）也称为用户态（user mode），是一般用户程序运行时所处的状态。处理器在处于目态时只能执行硬件机器指令的一个子集，即非特权指令。一旦用户程序在目态下执行特权指令，硬件将产生中断，进入操作系统，特权指令的执行将被制止。由于“置程序状态字”为特权指令，目态程序不能将其运行状态转换为管态，这样就可以防止用户有意或无意地侵入系统，从而起到系统保护的作用。

早期的微型计算机面向个人应用，不区分处理器状态，在这种环境中很难建立可靠的系统。现代计算机不仅改正了原有的设计，而且留有冗余，所支持的保护级别多达 5 级，即 R_0～R_4，其中 R_0 的权限最高，R_4 的权限最低。不过目前建立在计算机上的操作系统都只用了 R_0 和 R_4 这两个状态，操作系统运行于 R_0，用户程序运行于 R_4。

2. 状态转换

在系统运行的过程中，处理器状态处于动态变化之中，即时而处于目态，时而处于管态，这种变化是有规律的。

（1）目态到管态的转换

由于修改处理器状态字指令属于特权指令，只能在管态下执行，因而目态程序无法直接控制处理器状态的转换。处理器状态由目态转换为管态的唯一途径是中断。中断发生时，中断向量中的处理器状态字应标识处于管态，这个标识一般是由操作系统初始化程序来设置的。

（2）管态到目态的转换

管态到目态的转换可以通过修改程序状态字（置 PSW）来实现。由于操作系统运行于管态，用户程序运行于目态，因而这种状态转换伴随着由操作系统到用户程序的转换。

1.5.6 地址映射机构

在多道程序系统中，内存中同时存在多个程序，一个程序在内存中的存放位置是随机确定

的，而且通常可以改变，因此程序不能采用物理地址，而只能采用逻辑地址。为使每个程序的基本单位都能从 0 开始编址，硬件需要提供地址映射机构，负责将运行程序所产生的逻辑地址转换为内存物理地址。地址映射机构在不同的硬件环境中不尽相同，它在较大程度上决定了存储管理方式。

1.5.7 存储保护设施

在多道程序系统中，一个程序有意或无意产生的错误地址可能会侵犯其他程序空间甚至操作系统空间，一个程序对公共存储区域试图执行非法操作，这些都有可能影响其他程序甚至整个系统。为防止这些情况的发生，硬件必须提供存储保护设施，当发生存储越界错误或非法存取错误时，硬件的存储保护设施能够立即发现，并触发中断进入管态加以制止。

1.5.8 中断装置

发现并响应中断的硬件机构称为中断装置，中断装置具有以下两个功能。

① 发现中断。中断发生时能够识别。有多个中断事件同时发生时，按优先级别响应最高者。

② 响应中断。将目前运行进程的中断向量 PSW 和 PC 压入系统栈，然后根据中断原因到指定的内存单元将新的中断向量取出并送到寄存器中，从而控制转到相应的中断处理程序。

1.5.9 通道与 DMA 控制器

为使处理器从繁重的输入输出操作中解脱出来，同时为了增加处理器与设备之间、设备与设备之间的并行度，硬件提供了通道。通道是专门负责输入输出操作的处理器，具有自己的指令系统，可以执行通道程序，完成 CPU 委托的输入输出操作任务。

DMA（direct memory access，直接存储器存取）是与通道相似的输入输出方式，DMA 控制器接受 CPU 的委托完成数据在内存与块型设备之间的传输。与通道相比，DMA 控制器相对简单，没有专门的指令系统，一般一次只能传输一个数据块。

1.6 操作系统的界面形式


操作系统界面形式

从虚拟机的观点来看，操作系统是对计算机硬件的第一级扩充，也是最基本、最重要的扩充，配有操作系统的计算机在功能等方面与裸机相比大大增强了。在大多数情况下，用户通过操作系统与计算机硬件打交道，而不直接使用计算机硬件。那么，用户是如何使用操作系统的呢？换句话说，用户与操作系统之间的界面形式是什么样的呢？一般来说，操作系统为用户提供 5 种界面形式，即交互终端命令、图形用户界面、触屏用户界面、作业控制语言和系统调用命令。下面分别加以叙述。

1.6.1 交互终端命令

交互终端命令（又称命令行，command line）是分时操作系统所具有的界面形式。系统为交互终端用户提供一组交互式命令，用户可以通过终端键盘输入这些命令。每个输入命令都被操作系统中的命令解释程序所接收，该程序分析接收到的命令，然后调用操作系统中的相应模块完成此命令所要求的功能，最后将此命令的执行结果输出给用户，用户根据此结果决定下一条命令的输入，直到用户完成自己的工作。

交互终端命令界面由命令解释程序提供，该程序通常属于操作系统内核，但是 UNIX 系统的交互式命令解释程序由 shell（外壳）提供，而 shell 并不属于系统核心，而是运行于核心之外的目态程序，它通过系统调用与核心打交道，完成命令所要求的动作。

交互终端命令的一般形式为“命令名　选项　参数”，其中命令名指定操作功能，选项是对命令功能的调整，参数是命令操作的对象。一般系统都提供几十条甚至上百条交互式命令，操作人员必须熟记这些命令，才能对系统应用自如，这一般只有计算机专业人员才能做到。

1.6.2 图形用户界面

考虑用户尤其是非计算机专业人员使用计算机系统的方便性，现代操作系统都提供了图形用户界面（GUI）形式，GUI 在本质上也属于交互式界面形式，只不过界面由命令行转变为图形提示和鼠标操作。图形用户界面一般由视窗、图标、菜单、对话框等基本元素以及对基本元素所能进行的操作构成。在有些系统如 Windows 中，仍然保持一个行式命令的界面，不过该界面实际上是作为一个特殊的视窗实现的。

1.6.3 触屏用户界面

由于感应式接触屏幕的出现和普及，在手机和平板电脑上出现了触屏界面形式，这类设备一般没有键盘和鼠标，用户通过触摸和手的姿势产生中断，与操作系统进行交互，这类设备也可以在屏幕上仿真一个键盘，通过触摸虚拟按键输入文字。

1.6.4 作业控制语言

这是批处理系统所具有的界面形式。系统为用户提供一种作业控制语言（job control language，JCL）。当用户欲提交批作业时，他使用这种语言书写作业说明书，该说明书以操作系统所能识别的形式描述一个用户作业的处理步骤，然后将此说明书与程序、数据一道提交给操作系统。操作系统将按照作业说明书所规定的步骤一步一步地处理作业。

作业说明书是用一种专门的语言书写的，称为作业控制语言，作业控制语言是与操作系统相关的。不同的操作系统具有不同的作业控制语言，一般包含几十个作业控制命令。作业控制

语言是批处理系统中操作系统与用户之间的主要界面形式。

1.6.5 系统调用命令

系统调用（system call）命令也称为应用程序接口（application program interface，API），这是在用户程序级别上与操作系统打交道的方式。几乎所有类型的操作系统都有这种接口。操作系统为用户提供一组系统调用命令，用户可以将这些系统调用命令写在程序中。当用户程序在运行过程中执行到这些系统调用命令时，将发生自愿性中断，进入操作系统。操作系统将根据不同的系统调用命令转到相应的处理程序完成该调用命令所要求的服务。

系统调用命令通常可以分为以下几类：与文件相关的系统调用命令，如建立文件、撤销文件、打开文件、关闭文件、读写文件等；与进程相关的系统调用命令，如创建子进程、撤销子进程、跟踪子进程等；与进程间通信相关的系统调用命令，如发送消息、接收消息、发送信件、接收信件等；与资源相关的系统调用命令，如申请资源、释放资源等。

应当指出，用户与操作系统之间的 5 种界面形式是操作系统所提供的。对于不同的操作系统来说，这 5 种界面形式不尽相同。从用户的角度来说，当然希望能够将所有操作系统与用户之间的 5 种界面形式统一化、标准化。但是由于历史的原因以及机器结构等方面的差异，目前尽管在统一作业控制语言方面已经有人做了一些有益的工作，但是从整体来看进展不大。

1.7 操作系统的运行机理

操作系统是中断驱动的，考虑一个系统中并发执行的两个程序 P_1 和 P_2，假设在时刻 t_1 程序 P_1 执行，在时刻 t_2 程序 P_2 执行，$t_1<t_2$，则在时刻（t_1, t_2）之间一定发生过中断，即中断是程序切换的必要条件。

实际上，程序 P_1 不可能将 CPU 的使用权限直接交给程序 P_2，将处理器的使用权限由 P_1 转交给 P_2 只能由操作系统完成，而操作系统要完成 CPU 的重新分配必须首先获得 CPU 的使用权。操作系统取代 P_1 成为处理器的持有者的唯一途径是中断。中断将引出新的程序状态字并导致系统由目态转到管态，即进入操作系统。操作系统在执行完中断例程之后，既可能继续执行 P_1，也可能决定执行 P_2，这与 CPU 的调度原则有关。在后一种情况下，操作系统将保存 P_1 的状态信息，然后恢复 P_2 的状态信息并通过置程序状态字指令使系统转到目态运行 P_2。多道程序运行机理如图 1-12 所示。

有许多可能引起中断的事件，这些事件既可能与运行进程有关，如访管、地址越界、非法指令、溢出等，也可能与运行进程无关，如系统时钟、I/O 设备完成信号等。这里统一使用“中断”这一术语，在有些书籍中除“中断”外还使用“访管”“自陷”“功能调用”等术语，实质上它们都属于自愿性中断。

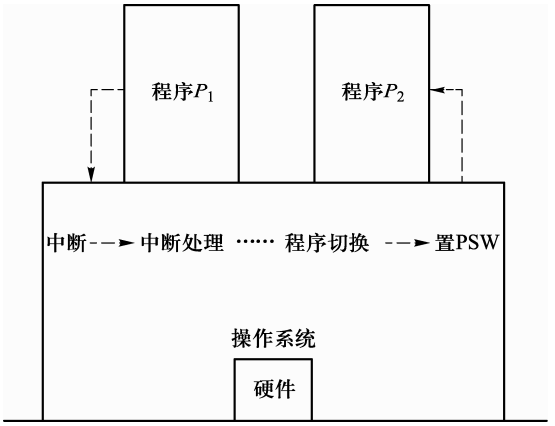

图 1-12 多道程序运行机理

1.8 研究操作系统的几种观点

操作系统也是一个程序，这个程序具有庞大、复杂的特点。为了深刻认识和描述操作系统，人们提出了研究操作系统的几种观点，包括进程观点、资源管理观点、虚拟机观点。这些观点彼此并不矛盾，而是站在不同的角度看待操作系统，每种观点都有助于理解和分析操作系统。

1.8.1 进程观点

关于什么是进程，在后面的章节中将给出详细描述，这里可以把进程理解为执行中的程序。毫无疑问，进程是操作系统乃至并发程序中最重要的概念。

进程观点将操作系统看成由若干个可以独立运行的程序和一个对这些程序进行协调管理的核心组成，这些运行的程序称为进程，每个进程完成某一特定任务。同时运行的进程之间可能会发生相互作用，这些相互作用表现为互斥、同步、通信、死锁、饥饿等，这些都是操作系统乃至并发程序设计中的核心问题。操作系统必须协调进程之间的相互作用，以使各个进程正常结束并得到正确的执行结果。

1.8.2 资源管理观点

计算机系统中配备有多种软硬件资源，如 CPU、内存、设备、文件等。这些资源通常是独占型的，即一次只能分配给一个请求者。并发执行的进程在运行过程中会使用这些资源，这些使用命令可能发生冲突，例如两个进程同时申请某一独占型设备。为此，必须有一个协调者负

责管理这些资源，这个管理者就是操作系统。

这种观点认为操作系统是一个资源管理程序，所管理的资源包括硬件资源（处理器、内存、设备等）和软件资源（文件、数据等），用户在使用资源前需要向操作系统提出申请，用完后将资源归还给操作系统。为了管理这些资源，操作系统需要给出这些资源的状态描述，并基于资源状态描述给出相应的资源管理程序。在多道程序系统中，可能有多个进程同时提出对同一种资源的使用申请，操作系统需要按某种策略满足这些请求。这种策略应当是公平的，同时也应当是高效的，这些将在资源管理的相关章节中详细介绍。

1.8.3 虚拟机观点

这种观点认为操作系统是一个虚拟机，由图 1-1 和图 1-2 可以看出，操作系统是硬件上运行的第一层系统软件，它对硬件功能进行第一次扩充。扩充之后的计算机系统具有功能强大、使用方便等特点。

虚拟机的思想在操作系统设计中有许多体现，例如可以在共享型设备和独占型设备的基础上构造虚拟设备，利用内存和外存储器实现虚拟存储，利用分时技术将一个处理器改造成多个虚拟处理器。

1.9 系统举例

1.9.1 Linux 系统

Linux 是 1991 年由 Linus Torvalds 主持开发的遵循 POSIX 标准的多用户、多任务操作系统，提供与 UNIX 兼容的应用程序界面，但是内核代码则完全重写。与 UNIX 不同，Linux 是一个源代码开放的自由软件，其发展得到众多 Linux 用户的协力支持。

1991 年，Linux 0.01 版运行于 Intel 80386 上，仅支持 Minix 文件系统，支持有限的设备驱动程序，不支持网络。1994 年推出的 Linux 1.0 版支持 UNIX 标准 TCP/IP 协议、与 BSD 兼容的套接字网络通信协议、增强的文件系统、SCSI 控制器对文件的高效访问，以及其他设备驱动程序。1995 年的 1.2 版是最后一个仅在 PC 平台上运行的 Linux。1996 年推出 Linux 2.0 版，可运行于多种平台上，支持对称多处器机制，同时增强了存储管理功能，支持核心级别线程、模块动态链接等，可以运行于 Sun SPARC、Power Mac 等硬件平台上。

Linux 具有以下特点：源代码开放，免费；系统稳定、可靠；速度快，效率高；内核模块化程度高，允许第三方配置文件系统及设备管理程序；功能完善；具有网络支持优势；标准化程度高。

1.9.2 Windows 10 系统

Windows 10 是基于 NT 技术构建的面向个人计算机平台的操作系统，本质上属于单用户系统，但可以组网并提供网络服务。系统具有如下特点。

① 具有多任务（包括多进程、多线程）管理功能，支持对称多处理，操作系统进程可以在任何可获得的处理器上运行，同一进程中的多个线程可以在不同处理器上同时运行。

② 支持客户-服务器计算模式，一台 PC 或工作站与一个主系统合作完成特定的服务程序。远程过程调用和本地过程调用是通用的服务机制。

③ 尽管不是纯的面向对象操作系统，但在设计上大量采用了面向对象思想，提供友好的图形操作界面。

④ Windows NT 的最初设计是微内核化的，但出于效率方面的考虑，目前的 Windows10 已经不是“纯”的微内核结构，许多系统服务功能已被放入核心。

习 题 一

1. 什么是操作系统？操作系统有哪些特性？

2. 硬件将处理器状态划分为两种，即管态和目态，这样做会给操作系统的设计带来什么好处?

3. 何谓特权指令?试举例说明。如果允许用户进程执行特权指令，会带来什么后果?举例说明。

4. 中断向量在计算机中的存储位置是由硬件决定的，还是由软件决定的?

5. 中断向量的内容是由操作系统程序决定的，还是由用户程序决定的?

6. 中断向量内的处理器状态字应当标明管态还是目态？为什么？

7. 系统如何由目态转换为管态？如何由管态转换为目态？

8. 中断与程序并发之间的关系是什么？

9. 根据用途说明“栈”和“堆”的差别。

10. 何谓系统栈？何谓用户栈？系统栈有何用途？用户栈有何用途?

11. 为何无法确定用户堆栈段的长度？

12. 为何堆栈段的动态扩充可能导致进程空间的变迁?

13. 何谓并行？何谓并发？在单处理器系统中，下述并行和并发现象哪些可能发生，哪些不会发生?

（1）进程与进程之间的并行　　（2）进程与进程之间的并发

（3）处理器与设备之间的并行　　（4）处理器与通道之间的并行

（5）通道与通道之间的并行　　（6）设备与设备之间的并行

14. 何谓作业？它包括哪几个部分？各个部分的用途是什么？

15. 试述批处理操作系统与分时操作系统的差别。

16. 从透明性和资源共享两个方面说明网络操作系统与分布式操作系统之间的差别。

17. 为什么构成分布式系统的主机一般都是相同的或兼容的?

18. 集群系统与分布式系统有何差别?

19. 何谓云存储? 何谓云计算?

20. 为什么嵌入式操作系统通常采用微内核结构? 微内核结构包括哪些内容?

21. 微内核结构有哪些优点和缺点?

22. 操作系统为用户和上层软件提供哪种界面形式? 相应的界面形式适用于哪种应用环境?