

# 第一章作业

## 一、操作系统的作用是什么？

### 1、OS 作为用户与计算机硬件系统之间的接口

OS 处于用户与计算机硬件系统之间，用户通过 OS 来使用计算机系统。或者说，用户在 OS 帮助下能够方便、快捷、可靠地操纵计算机硬件和运行自己的程序。

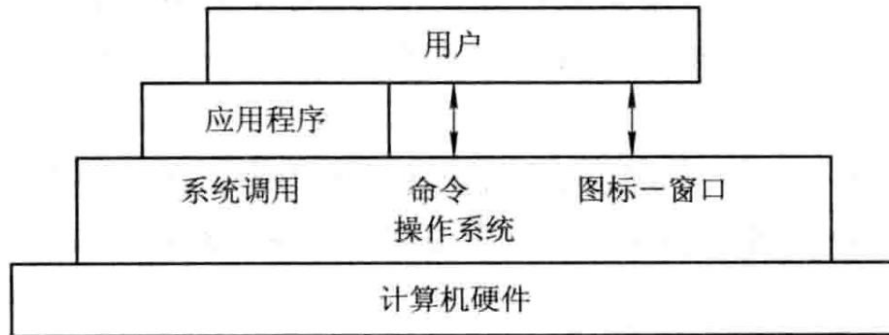


图 1-1 OS 作为接口的示意图

### 2、OS 作为计算机系统资源的管理者

在一个计算机系统中，通常都会有多种硬件和软件资源。归纳起来可将这些资源分为四类：处理机、存储器、I/O 设备以及文件（数据和程序）。相应地，OS 的主要功能也是要对这四类资源进行有效的管理。处理机管理是用于分配和控制处理机；存储器管理主要负责内存的分配与回收；I/O 设备管理是负责 I/O 设备的分配（回收）与操纵；文件管理是用于实现对文件的存取、共享和保护。可见，OS 的确是计算机系统资源的管理者。

除此之外，当一台计算机系统同时供多个用户使用，诸多用户对系统中共享资源的需求（包括数量和时间）有可能发生冲突。为此，操作系统必须对使用资源的请求进行授权，以协调各用户对共享资源的使用。

### 3、OS 实现了对计算机资源的抽象

对于一台完全无软件的计算机系统（即裸机），由于它向用户提供的仅是硬件接口（物理接口），因此，用户必须对物理接口的实现细节有充分的了解，这就致使该物理机器难于逐级使用。为了方便用户使用 I/O 设备，人们在裸机上覆建立一层 I/O 设备管理软件，如图 1-2 所示，它来实现对 I/O 设备操作的细节，并向上将 I/O 设备抽象为一组数据结构以及一组 I/O 操作命令，如 read 和 write 命令，这样用户可以用这些数据结构及操作命令来进行数据输入或输出，而无需关心 I/O 是如何具体实现的。此时用户所看到的机器是一台比裸机功能更强、使用更方便的机器。换言之，在裸机上输出的 I/O 软件隐藏了 I/O 设备的具体细节，向上提供了一组抽象的接口。

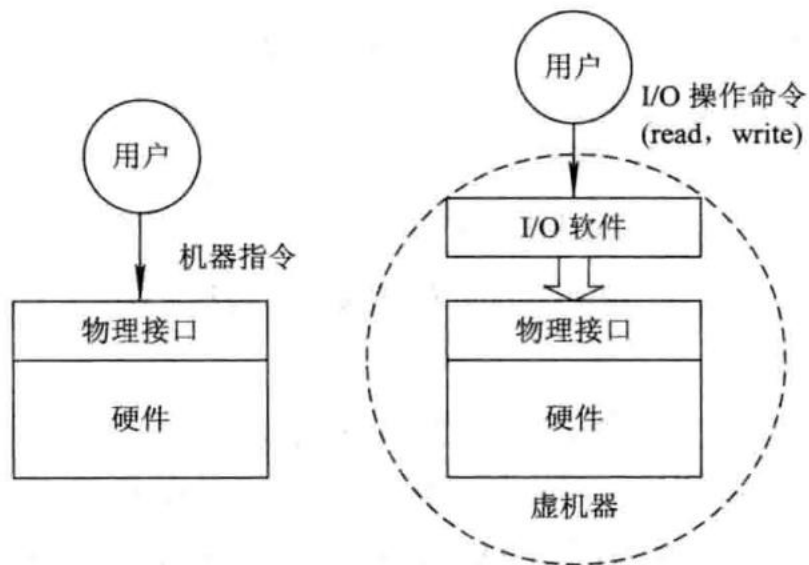


图 1-2 I/O 软件隐藏了 I/O 操作实现的细节

通常把覆盖了上述软件的机器称为扩充机器或虚拟机。它向用户提供了一个对硬件操作的抽象模型。用户可利用该模型提供的接口使用计算机，无需了解物理接口实现的细节，从而使用起来更容易地使用计算机硬件资源。亦即，I/O 设备管理软件实现了对计算机硬件的第一个层次的抽象。

同理，为了方便用户使用文件系统，又可在第一层软件（I/O 管理软件）上再覆盖一层用于文件管理的软件，由它来实现对文件操作的细节，并向上层提供一组实现对文件进行存取操作的数据结构及命令。这样，用户可利用该软件提供的数据结构及命令对文件进行存取。此时用户所看到的是一台功能更强、使用更方便的虚拟机。亦即，文件管理软件实现了对硬件资源操作的第二个层次的抽象。依此类推，如果在文件管理软件上再覆盖一层面面向用户的窗口软件，则用户便可在窗口环境下方便地使用计算机，从而形成一台功能更强的虚拟机器。

由此可知，OS 是铺设在计算机硬件上的多层软件的集合，它们不仅增强了系统的功能，还隐藏了对硬件操作的具体细节，实现了对计算机硬件操作的多个层次的抽象模型。值得说明的是，不仅可在底层对一个硬件资源加以抽象，还可在高层对该资源底层已抽象的模型再次进行抽象，成为更高层的抽象模型。随着抽象层次的提高，抽象接口所提供的功能就越强，用户使用起来也越方便。

## 二、操作系统具有哪几个特征？它们之间有何关系？

### 1、并发性

操作系统支持多个任务（程序或进程）同时运行，这些任务可以并发地执行。并发性是操作系统最重要的特征之一，它允许多个进程共享 CPU 和其他系统资源，使系统能够同时处理多个任务，提高系统的资源利用率和吞吐量。

### 2、共享性

操作系统允许多个用户或多个程序共享计算机系统硬件资源和软件资源。例如，多个进程可以同时使用 CPU、内存、I/O 设备等硬件资源，也可以访问同一文件系统或数据库。共享性要求操作系统能够合理地管理和分配资源，确保共享过程中数据的安全性和完整性。目前主要实现资源共享的方式有以下两种：

#### 1. 互斥共享方式

系统中的某些资源，如打印机、磁带机等，虽然可以提供给多个进程（线程）使用，但应规定在一段时间内，只允许一个进程访问该资源。为此，在系统中应建立一种机制，以保证多个进程对这类资源的互斥访问。

当进程 A 要访问某资源时，必须先提出请求。若此时该资源空闲，系统便可将之分配给请求进程 A 使用。此后若再有其它进程也要访问该资源，只要 A 未用完就必须等待。仅当 A 进程访问完并释放系统资源后，才允许另一个进程对该资源进行访问。这种资源共享方式称为互斥共享。把这种在一段时间内只允许一个进程访问的资源，称为临界资源（或独占资源）。系统中的大多数物理设备，以及栈、变量和表格，都属于临界资源，都只能被互斥地共享。为此，在系统中必须配置某种机制，用于保证诸进程互斥地使用临界资源。

## 2.同时访问方式

系统中还有另一类资源，允许在一段时间内由多个进程“同时”对它们进行访问。这里所谓的“同时”，在单处理机环境下是宏观意义上的，而在微观上，这些进程对该资源的访问是交替进行的。典型的可供多个进程“同时”访问的资源是磁盘设备。某些用重入编码写的文件也可以被“同时”共享，即允许若干个用户同时访问该文件。

并发和共享是多用户（多任务）OS 的两个最基本的特征。它们又是互为存在的条件。即一方面资源共享是以进程的并发执行为条件的，若系统不允许并发执行也就不存在资源共享问题；另一方面，若系统不能对资源共享实施有效管理，以协调各诸进程对共享资源的访问，也必然会影响到诸进程的并发执行的程度，甚至根本无法并发执行。

## 3、虚拟性

操作系统通过虚拟化技术为用户提供虚拟资源，使得有限的物理资源看起来是无限的。在操作系统中虚拟技术分为时分复用技术和空分复用技术：

### 1. 时分复用技术

时分复用技术是通过时间的分割来共享系统资源的一种方式。具体来说，它将时间划分成若干个时间片，每个时间片分配给不同的进程或任务。每个进程在它的时间片内独占使用 CPU，时间片结束后，系统切换到下一个进程。

### 2. 空分复用技术

空分复用技术是通过空间的划分来共享系统资源的一种方式。它将系统资源（如内存、存储空间）分割成多个部分，每个部分分配给不同的用户或进程，多个用户或进程可以同时使用这些分配的资源。

## 4、异步性

操作系统支持异步事件处理，即系统中的进程或任务可以在任何时候被打断，操作系统会根据中断的优先级和类型，进行相应的调度和处理。这使得系统能够及时响应外部事件（如用户输入、硬件中断等），保持系统的灵活性和实时性。

## 关系：

并发性和资源共享性是操作系统最基本的特征。为了提高计算机资源的利用率，操作系统必须采用多道程序设计技术，使多个程序能够共享系统的资源并并发执行。资源共享的实现依赖于程序的并发执行，而并发执行的有效性也依赖于对资源共享的有效管理。

虚拟性以并发性和资源共享性为前提。为了使并发进程能够更方便、更有效地共享资源，操作系统常采用多种虚拟技术在逻辑上增加 CPU 和设备的数量以及存储器的容量。这解决了众多并发进程对有限系统资源的争用问题。异步性是并发性和资源共享性的必然结果。操作系统允许多个并发进程共享资源、相互合作，这使得每个进程的运行过程受到其他进程的制约，不再“一气呵成”，从而产生了异步性特征。

### 三、试述多道程序设计技术的基本思想。为什么采用多道程序设计技术可以提高资源利用率？

在该系统中，用户所提交的作业先存放在外存上，并排成一个队列，称为“后备队列”。然后由作业调度程序按一定的算法，从后备队列中选择若干个作业调入内存，使它们共享 CPU 和系统中的各种资源。由于同时在内存中装有若干道程序，这样便可以在运行程序 A 时，利用其因 I/O 操作而暂停执行时的 CPU 空档时间，再调度另一道程序 B 运行，同样可以利用程序 B 在 I/O 操作时的 CPU 空档时间，再调度程序 C 运行，使多道程序交替地运行，这样便可以保持 CPU 处于忙碌状态。

多道程序设计允许在一个程序进行 I/O 操作期间，切换到其他程序执行，使 CPU 不必等待 I/O 操作结束，这样大大减少了 CPU 的空闲时间，保持 CPU 处于忙碌状态。并且多道程序设计可以在内存中装入多道程序以提高内存的利用率；还可以提高 I/O 设备的利用率。

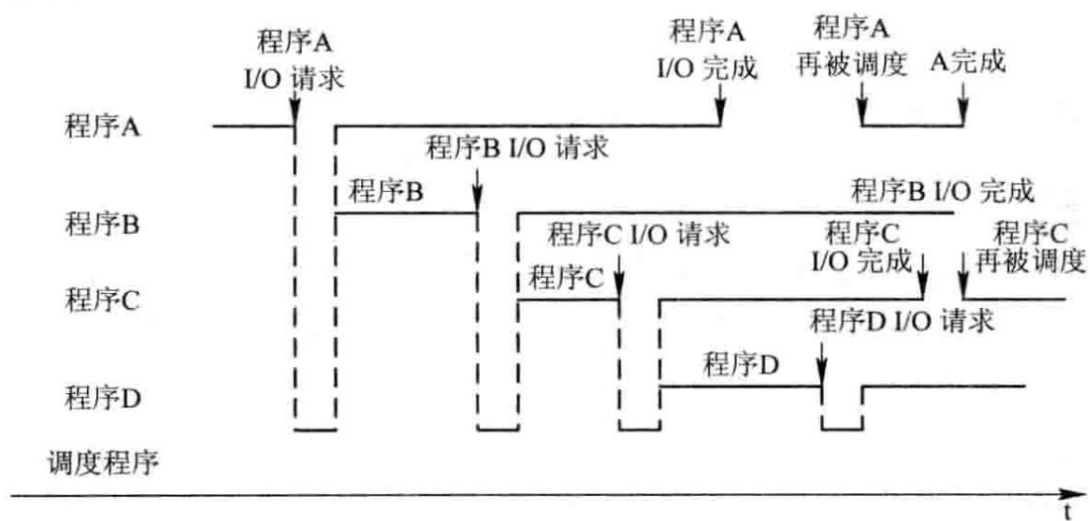


图 1-6 多道程序的运行情况

### 四、什么是分时系统？其主要特征是什么？适用于哪些应用？

分时系统是指，在一台主机上连接了多个配有显示器和键盘的终端并由此组成的系统，该系统允许多个用户同时通过自己的终端，以交互方式使用计算机，共享主机中的资源。

分时系统的主要特征有以下四点：

(1) 多路性。该特性是指系统允许将多台终端同时连接到一台主机上，并按分时原则为每个用户服务。多路性允许多个用户共享一台计算机，显著地提高了资源利用率，降低了使用费用，从而促进了计算机更广泛的应用。

(2) 独立性。该特性是指系统提供了这样的用机环境，即每个用户在各自的终端上进行操作，彼此之间互不干扰，给用户的感觉就像是他一人独占主机进行操作。

(3) 及时性。及时性是指用户的请求能在很短时间内获得响应。这一时间间隔是根据人们所能接受的等待时间而确定的，通常仅为 1~3 秒钟。

(4) 交互性。交互性是指用户通过终端与系统进行广泛的人机对话。其广泛性表现在：用户可以请求系统提供多方面的服务，如进行文件编辑和数据处理，访问系统中的文件系统和数据库系统，请求提供打印服务等。

分时系统主要适用于需要多用户交互、高响应速度以及资源共享的应用场景，例如：

(1) 多用户环境：在需要支持多个用户同时使用计算机资源的场景中，分时系统非常适用。例如，在大学、研究机构或企业内部，多名用户需要通过终端共享同一台主机进行程序设计、数据处理或文档编辑等操作。

(2) 交互式应用程序：分时系统适合需要用户与计算机频繁交互的应用程序，如文本编辑器、编程环境、数据库查询和管理系统等。用户能够在短时间内获得响应，这对于实时交互的用户体验非常重要。

(3) 在线服务系统：包括早期的电子邮件系统、在线编程系统、计算机辅助教学（CAI）系统、网络论坛等，这些系统需要支持多个用户同时进行操作，而分时系统能够高效地管理这些并发请求。

(4) 远程访问与控制：在远程终端登录和操作主机的情况下，例如远程文件管理、远程数据分析或远程调试程序等，分时系统能够通过合理的时间片分配和资源管理，使得每个远程用户的操作看起来像是独占使用计算机。

(5) 开发与测试环境：在软件开发和测试过程中，多个开发人员需要同时进行编程、测试和调试，而分时系统可以提供一个共享的开发环境，确保资源的有效利用并降低成本。