

第1章 操作系统引论

1.1 操作系统概述

一、操作系统的概念、特征、功能和提供的服务

1. 操作系统的概念

操作系统介于计算机硬件系统和其他所有软件系统之间，为所有软件系统使用计算机硬件提供各项服务。操作系统是一组控制和管理计算机硬件和软件资源、合理地组织计算机工作流程以及方便用户的程序集合。

操作系统追求的基本目标是用户使用计算机的方便性，以及计算机硬件的高效率运行。操作系统主要有两方面重要的作用。

(1) 操作系统管理系统中的各种资源，包括硬件及软件资源。

在计算机系统中，所有硬件部件（如 CPU、存储器和输入/输出设备等）均称作硬件资源，而程序和数据等信息称作软件资源。因此，从微观上看，使用计算机系统就是使用各种硬件资源和软件资源。特别是在多用户和多道程序的系统中，同时有多个程序在运行，这些程序在执行的过程中可能会要求使用系统中的各种资源。操作系统就是资源的管理者和仲裁者，由它负责在各个程序之间调度和分配资源，保证系统中的各种资源得以有效地利用。

(2) 操作系统要为用户提供的良好界面。

一般来说，使用操作系统的用户有两类：一类是最终用户，另一类是系统用户。最终用户只关心自己的应用需求是否被满足，而不在意其他情况。至于操作系统的效率是否高，所有的计算机设备是否正常，只要不影响他们的使用，他们则一律不去关心，而后面这些问题则是系统用户所关心的。

操作系统必须为最终用户和系统用户这两类用户的各种工作提供良好的界面，以方便用户的工作。典型的操作系统界面有两类：一类是命令行界面，如 UNIX、Linux 等；另一类则是图形化的操作系统界面，如 Windows、Linux 等。

2. 操作系统特征

多道程序设计技术可以极大地提高计算机资源的利用率，但它也改变了程序的工作环境，使程序由顺序执行变成并发执行，因此带来一些新的复杂问题，使得现代操作系统具有如下一些特征：

(1) 并发性 (Concurrency)。这种标题要缩进两个字

并发性是指两个或多个事件在同一时间间隔内发生；并行性是指两个或多个事件在同一时刻发生。在多道程序环境下，并发性是指宏观上在一段时间内有多道程序在同时执行。但在单处理机系统中，每一个时刻仅能执行一道程序，微观上，这些程序是在交替执行。

(2) 共享性 (Sharing)。

共享是指系统中的所有资源（如 CPU、内存、I/O 设备及软件资源）不再为一个程序所独占，而是供同时存在于系统中的多道程序共同使用。根据资源属性不同，可有互斥共享和同时共享两种不同的共享方式。

(3) 虚拟性 (Virtual)。

虚拟是指通过某种技术把一个物理实体变成若干个逻辑上的对应物。物理实体是实际存

在的，是实的；逻辑物体是用户感觉到的，是虚拟的。例如在单 CPU 多道分时系统中，通过多道程序技术和分时技术可以把一个物理 CPU 虚拟为多台逻辑上的 CPU，使每个终端用户都认为有一台“独立”的 CPU 为它运行，用户感觉的 CPU 是虚拟 CPU。

(4) 异步性 (Asynchronism)。

在多道程序环境下，允许多个程序并发执行，但由于资源等因素的限制，程序的执行是以“走走停停”的方式运行，即程序是以异步方式运行的。

3. 操作系统功能

操作系统作为系统的资源管理者，并作为计算机和用户间的接口，它的主要功能有：

(1) 处理机管理。

处理机管理的主要任务是对处理机进行分配，并对其运行进行有效的控制和管理。在多道程序环境下，处理机的分配以进程为基本单位，运行以进程或线程为基本单位，因而对处理机管理可归纳为对进程的管理。进程管理包括进程控制、进程同步、进程通信和调度等。

(2) 存储管理。

存储管理的主要任务是为多道程序的运行提供良好的环境，方便用户使用存储器，提高存储器的利用率，以及能从逻辑上来扩充内存，为此存储管理应具有内存分配、内存保护、地址映射和虚拟存储器等功能。

(3) 文件系统。

计算机系统的软件信息都以文件形式进行管理，操作系统中负责此任务的部分是文件系统，文件系统的任务是对用户文件和系统文件进行管理，以方便用户使用，并保证文件的安全性，为此文件系统管理应具有对文件存储空间的管理、目录管理、文件共享和保护等功能。

(4) 设备管理。

设备管理的任务是登记各 I/O 设备状态，管理并完成用户提出的 I/O 请求，按一定的策略为用户分配 I/O 设备。同时提高 CPU 和 I/O 设备的利用率，提高 I/O 速度，方便用户使用 I/O 设备，为此设备管理应具有缓冲器管理、设备分配、设备处理及虚拟设备等功能。

(5) 用户接口。

为了方便用户使用操作系统，操作系统向用户提供了“用户与操作系统的接口”，操作系统接口分成两类：第一类是命令接口，它提供一组键盘和鼠标命令，供用户去组织和控制程序的运行。第二类是程序级接口，它提供一组系统调用供其它程序调用。

4. 操作系统提供的服务

操作系统要为用户程序的执行提供一个良好的运行环境，它为程序和用户提供各种服务，当然不同的操作系统提供的服务不完全相同，但有许多是共同的。操作系统提供给程序和用户的共性服务大致如下：

(1) 程序执行：将用户程序和数据装入内存，为其运行做好一切准备工作并启动它执行。当程序编译或运行执行出现异常时，应能报告发生的情况，终止程序执行或进行适当处理。

(2) I/O 操作：程序运行过程中需要 I/O 设备上的数据时，可以通过 I/O 命令或 I/O 指令，请求操作系统的服务。操作系统不允许用户直接控制 I/O 设备，而能让用户以简单方式实现 I/O 控制和读写数据。

(3) 文件系统操纵：文件系统让用户按文件名来建立、读写、修改、删除文件，使用方便，安全可靠。当涉及多用户访问文件时，操作系统将提供信息保护机制。

(4) 通信：在多数情况下，一个进程要与另外的进程交换信息，这种通信发生在两种场合，一是在同一台计算机上执行的进程之间通信；二是在被网络连接在一起的不同计算机上执行的进程之间通信。进程通信可以借助共享内存实现，也可以使用消息传送技术

实现。

(5) 错误检测和处理：操作系统能捕捉和处理各种硬件或软件造成的差错或异常，并让这些差错或异常造成的影响缩小在最小的范围内，必要时及时报告给操作员或用户。

(6) 资源分配：多个用户或多道作业同时运行时，每一个必须获得系统资源。系统中的各类资源均由操作系统管理，如 CPU 时间、内存资源、文件存储空间等，都配有专门的分配程序，而其它资源(如 I/O 设备) 配有更为通用的申请与释放程序。

(7) 统计：人们希望知道用户使用计算机资源的情况，如用了多少？什么类型？以便用户简单地进行使用情况统计，可以作为进一步改进系统服务，对系统进行重组的有价值的工具。

(8) 保护：在多用户多任务计算机系统中，文件所有者能对其创建的文件进行控制使用，保护意味着要确保对系统资源的所有存取要受到控制。用户对各种资源的需求经常发生冲突，为此，操作系统必须做出合理的调度。

5. 系统调用

系统调用是操作系统提供服务的唯一途径。在最底层，系统调用允许运行程序直接向操作系统发出请求。在高层，命令解释程序接受用户发出的请求。命令可以来自文件（批处理模式），或者直接来自键盘输入（交互模式或分时模式）。进而，命令解释程序通过系统调用满足用户的操作要求。

在讨论操作系统如何使其系统调用可用之前，首先用一个例子来解释如何使用系统调用：编写一个从一个文件读取数据并复制到另一个文件的简单程序。程序首先所需要的输入是两个文件的名称：输入文件名和输出文件名。根据操作系统设计的不同，这些名称有许多不同的表示方法。一种方法是程序向用户提问然后得到两个文件名。对于交互系统，这种方法需要一系列的系统调用：先在屏幕上写出提示信息，再从键盘上读取定义两个文件名称的字符。对于基于鼠标和基于图标的系统，一个文件名的菜单通常显示在一个窗口中。用户通过鼠标选择源文件名，另一个类似窗口可以用来选择目的文件名。这个过程需要许多 I/O 系统调用。

在得到两个文件名后，该程序打开输入文件并创建输出文件。每个操作都需要另一个系统调用。每个操作都有可能遇到错误情况。当程序设法打开输入文件时，它可能发现该文件不存在或者该文件受保护而不能访问。在这些情况下，程序应该在终端上打印出消息（另一系列系统调用），并且非正常地终止（另一个系统调用）。如果输入文件存在，那么必须创建输出文件。用户可能会发现具有同一名称的输出文件已存在。这种情况可能导致程序中止（一个系统调用），或者必须删除现有文件（另一个系统调用）并创建新的文件（另一个系统调用）。对于交互式系统，另一选择是问用户（一系列的系统调用以输出提示信息并从终端读入响应）是否需要替换现有文件或中止程序。

现在两个文件都已设置好，可以进入循环以从输入文件中读（一个系统调用）并向输出文件中写（另一个系统调用）。每个读和写都必须返回一些关于各种可能错误的状态信息。对于输入，程序可能发现已经到达文件的结束，或者在读过程中发生了一个硬件失败（如奇偶检验误差）。对于写操作，根据输出设备的不同可能出现各种错误（如没有磁盘空间、打印机没纸等）。

最后，在整个文件复制完成后，程序可以关闭两个文件（另一个系统调用），在终端或窗口上写一个消息（更多系统调用），最后正常结束（最后的系统调用）。可见，一个简单的程序也会大量使用操作系统。通常，系统每秒执行数千个系统调用。图 1.1 显示了这个系统调用序列。

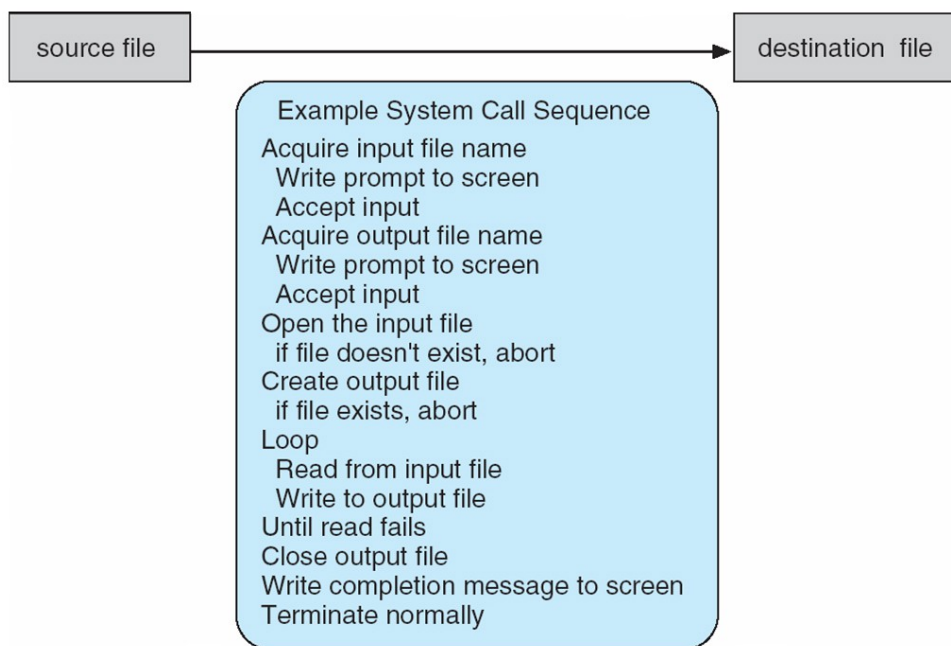


图 1.1 系统调用顺序例子

绝大多数程序设计语言的运行时支持系统（与编译器一起的预先构造的函数库）提供了系统调用接口，作为应用程序与操作系统的系统调用的链接。系统调用接口截取 API 的函数调用，并调用操作系统中相应的系统调用。通常，每个系统调用一个与其相关的数字，系统调用接口根据这些数字维护一个列表索引。然后，系统调用接口来调用所需的操作系统内核中的系统调用，并返回系统调用状态及其他返回值。

调用者不需要知道如何执行系统调用或者执行过程中它做了什么，它只需遵循 API 并了解执行系统调用后，系统做了什么。因此，对于程序员，通过 API 操作系统接口的绝大多数细节被隐藏起来，并被执行支持库所管理。API、系统调用接口和操作系统之间的关系如图 1.2 所示，它表现了操作系统如何处理一个调用 `open()` 系统调用的用户应用。

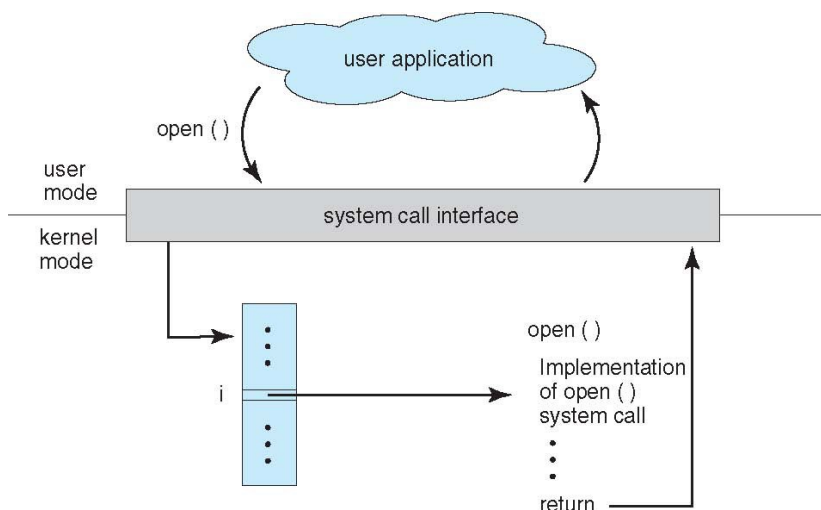


图 1.2 处理一个调用 `open()` 系统调用的用户应用程序

系统调用可分为五大类：进程控制、文件操纵、设备管理、信息维护和通信等。由命令解释程序或系统程序来完成的高级别请求需要转换成一系列的系统请求。系统服务可分成许多类型：程序控制、状态请求、和 I/O 请求。程序出错可作为对服务的一种隐式请求。

二、操作系统的发展与分类

操作系统已经发展了半个世纪，它始终围绕着两个主要目的。第一，操作系统试图调度计算活动以确保计算机系统的高性能。第二，操作系统提供一个便于开发和运行程序的环境。

最初，计算机只能通过控制台来使用。汇编程序、装入程序、连接程序和编译程序等持续改善用户编程的方便性，但代价是耗费大量的设置时间。这些设置时间可以与系统运行并行处理，即采用批处理方式，从而达到了改善系统性能的目的。

批处理系统通过驻留内存的操作系统，自动装入作业，使计算机不再需要等待人工操作。但是，CPU 使用率仍然很低，这是因为 I/O 设备的速度要比 CPU 慢。当然，可以对慢设备采取离线操作的方法，即在一个 CPU 上使用多个磁带读入和磁带打印系统。

为了改善计算机系统的整体性能，操作系统引入了多道程序设计概念，它成为现代操作系统的基本特征。在多道程序设计环境里，多个作业可以同时驻留内存；CPU 在这些作业之间来回切换以提高其使用率；此法也降低了执行作业所需要的总时间。

多道程序设计也允许分时。分时操作系统允许多个用户同时交互地使用一个计算机系统。

PC 时代的微机与大型机相比，相对比较小并且便宜。这些计算机的操作系统在许多方面都得益于大型机操作系统的发展成果。不过，由于单个用户可以独占计算机，因而 CPU 利用率不再是主要问题。因此，有的大型操作系统的设计决策不再适用于这些小系统。其他设计决策，如安全性等，因为 PC 可以通过网络或 WWW 与其他计算机和用户相连，对于微型机和大型机都同样适用。

并行系统有多个紧密耦合的 CPU。这些 CPU 共享总线，有时也共享内存和外设。并行系统提供了高计算量和高可靠性。

分布式系统允许对分布在各地的主机资源进行共享。集群系统允许多个系统对位于共享存储中的数据进行操作，即使一部分集群成员失败，也能正常工作。

硬实时操作系统具有明确的、固定的时间约束，处理必须在规定的约束内完成，常常用于控制专用应用设备。软实时系统没有严格的时间约束，不支持最终期限调度。

近来，由于 Internet 和 WWW 的影响，现代操作系统也集成和加强了 WWW 浏览器、网络和通信能力。