

# 第 1 篇

## 操作系统原理

操作系统是现代计算机系统中一种最重要的系统软件。一般情况下,当用户启动计算机后,直接面对操作系统,或面对在操作系统上运行的其他软件。因此,每个想要正确使用计算机系统的人,对操作系统都应该有所了解,对其中有些内容甚至应该相当熟悉。

计算机技术发展到今天,操作系统已经变得十分复杂。对大多数人来说,没有必要也不可能完全理解并掌握它,先学习一般原理,然后有重点地研究感兴趣的部分才是明智的做法。我们接触的一个个具体的操作系统,原理基本相同或者十分相似。学习一般原理,有助于我们熟练地使用计算机;理解操作系统技术发展的来龙去脉,就能自信地面对新的操作系统。

按传统说法,操作系统的主要工作是管理计算机系统中的处理器、存储器、外部设备和文件这四大资源。自从引入线程概念后,也有人仅将存储器、外部设备和文件称为资源,而将处理器另外分开。考虑到教材特点,本篇组织章节时也采用了这种划分方法。另外,虽然今天的操作系统还被要求管理网络、用户和安全,但是因篇幅所限不专门对此进行讨论。

本篇包含 3 章,第 1 章简要介绍操作系统的概貌,第 2、3 章分别介绍操作系统处理器管理和资源管理的基本原理。作业管理、进程管理和线程管理与处理器联系较紧密,因此归在处理器管理中;存储器管理、外部设备管理和文件管理则归在资源管理中。



# 操作系统概述

本章将介绍操作系统的定义、目标、功能、特征、类型和结构,简要介绍几种典型的操作系统,再介绍与操作系统紧密相关的硬件知识,最后介绍操作系统的用户界面。

## 1.1 操作系统的定义与目标

### 1.1.1 操作系统的地位

计算机系统能够按人的要求接收和存储信息,经过处理和计算,输出结果信息,靠的是硬件和软件两部分配合工作。硬件是指处理器、存储器、I/O 设备和通信装置等(如图 1-1 所示);软件是指为完成特定任务而由硬件执行的程序、数据和其他相关文档。

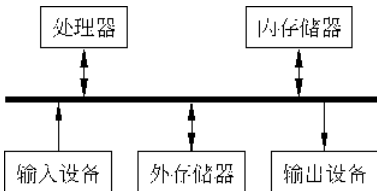


图 1-1 计算机硬件组成示意图

计算机的体系结构一直在发展变化,但占主流地位的仍是以存储程序原理为基础的冯·诺依曼(Von Neumann)型计算机。存储程序原理的基本点是:程序由指令组成,并与数据一起存放在计算机存储器中;机器一经启动,就能按照程序规定的逻辑顺序从存储器中读出指令逐条执行,自动完成程序所描述的工作。

计算机软件可以分为 3 大类:系统软件、支撑软件和应用软件。系统软件最靠近硬件,与具体应用领域无关,其他软件一般都通过系统软件发挥作用,操作系统和编译程序等都是系统软件;支撑软件用于支撑其他软件的开发和维护,典型的如软件开发环境,支撑软件 and 系统软件之间并没有严格的界限;应用软件则是针对特定应用问题的专用软件,例如字处理程序和管理信息系统等。

由系统软件和支撑软件组成的环境,为应用软件的开发和维护提供了方便;支撑软件和应用软件的运行一般都离不开系统软件的支持。现代计算机系统已不再被简单地认

为是一种电子设备,而是由硬件和软件的多个层次构成的十分复杂的整体。图 1-2 给出一般计算机系统的层次结构。操作系统在计算机系统中占据着非常重要的位置,它是紧靠着硬件的第一层软件。计算机硬件加载了操作系统,才成为可以协调运行的、使用方便的、高效的计算机系统。其他软件一般都必须运行在操作系统之上。

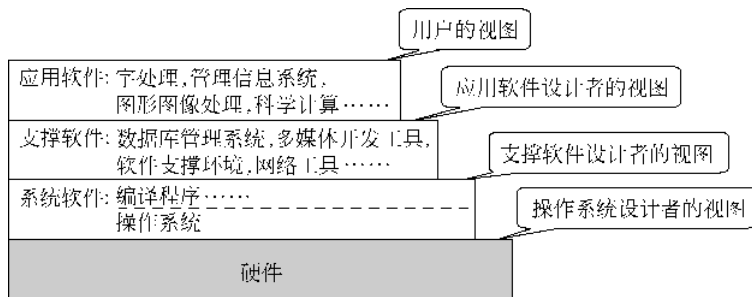


图 1-2 一般计算机系统的层次结构

在计算机系统中,中央处理器(CPU,有时也简称处理器)、存储器、I/O 设备等所有硬件称为硬件资源,程序和数据等称为软件资源。使用计算机系统就是使用这些资源。程序在系统中运行时,需要操作系统来进行程序调度和资源分配,以保证系统资源的有效利用。特别是多个用户、多个程序同时运行时,这种管理更显得必要,而且十分复杂。各种计算机都离不开操作系统,否则将变得非常难以使用。

### 1.1.2 操作系统的定义

操作系统可以定义为:操作系统由一组程序组成,这组程序能够有效地控制和管理计算机系统中的硬件和软件资源,合理地组织计算机系统的工作流程和控制程序的执行,使计算机系统能高效地运行,向用户提供各种服务,使用户能够灵活、方便、有效地使用计算机。

这个定义指出操作系统是一组程序,更确切地说,还包括一些专门的数据结构。定义中所要求的“有效”,主要指系统运行效率和资源利用率,尤其要尽可能提高处理器的利用率,其他的资源(例如内存和外存)则应该在保证访问效能的前提下尽可能减少浪费。“合理”主要是指要公平地对待各种用户程序,不让有些用户感到被歧视;恰当地解决争用资源引起的冲突,以保证系统不会因此而崩溃;巧妙地组织工作流程,以消除系统中“忙闲不均”现象。“方便”主要是对用户界面的要求,这包括人机交互接口和程序设计接口两方面的易用性、易学性和易维护性。

### 1.1.3 操作系统的目标

上述定义反映了操作系统应该达到的三个目标。

#### (1) 方便用户

操作系统有两类用户。一类是最终用户,他们关心自己的应用需求是否被满足;另一类是系统用户,他们监视着系统的整体运行状态,如空间的使用情况,是否发生通信堵塞,有无黑客攻击等。利用控制语言(或终端命令)和系统调用等方式,用户可以方便地使用操作系统提供的各种功能。

### (2) 提高效率

操作系统根据用户对其程序运行的控制意图和系统资源的使用情况,合理地组织系统的工作流程,高效地满足用户的要求。在操作系统出现前,用户程序只能低效地使用计算机系统资源。操作系统通过采用多道、中断、分时和缓冲等技术,实现了对系统资源的高效使用。

### (3) 管好资源

要方便用户和提高效率,就必须管好资源。操作系统对于系统中的资源实行统一管理、合理分配和及时回收。多个程序运行在同一个计算机系统上,它们的资源请求对系统来说具有动态性和随机性,程序往往因争用资源而“僵持不下”,这离不开操作系统的有效预防和及时解决。

方便用户、提高效率和管好资源这三个方面是密切相关的。管好资源有利于提高效率,也在一定程度上为用户提供了方便;提高效率能使更多的用户得到服务,使资源得到充分的利用;方便用户则能吸引更多的用户,为提高资源利用率创造更多机会。但是三者之间也有一定的制约,例如,为了方便用户有时需要牺牲一点效率,为了管好整个系统的资源就可能会使某些用户感觉不便。因此需要权衡利弊,统筹兼顾,对单一目标的追求要适可而止。

## 1.2 操作系统的功能与特征

在介绍操作系统的功能和特征之前,先了解一下操作系统的发展历史。

### 1.2.1 操作系统的形成

#### 1. 手工操作方式

早期的计算机系统通常由 CPU、内存储器 and 外部设备(包括卡片输入机、卡片穿孔机、打印机和磁带机等)连接而成。它只提供原始的功能,不带有任何帮助用户工作的程序。用户的程序和数据保存在外部存储介质上,计算机系统由单个用户独占使用。这种方式下,计算机系统各部分串行工作,造成了资源的浪费。而且手工操作还容易出错。

因此,人们设计了 I/O 控制程序。I/O 控制程序长期存储在计算机系统里,用户需要 I/O 时就发出相应的指令调用这些程序。I/O 控制程序成为计算机系统的一部分。此外,人们又在计算机系统中加进一些开关命令的解释程序,以使用户能够通过开关来控制程序的装入和运行。但是,用户仍然靠控制台上的指示灯、开关和按钮来控制计算机系统,基本上仍是一种手工操作方式,低效率和易出错是这种方式的致命弱点。

#### 2. 监控程序

20 世纪 60 年代初,随着 FORTRAN 和 ALGOL 等程序设计语言及其编译程序和程序库的出现,逐渐出现了监视和控制计算机硬件和软件的程序——监控程序。监控程序的任务是帮助操作员管理计算机的一部分资源,控制用户程序的执行,在一定程度上实现资源共享。通常,监控程序的主要部分是预装在计算机内的,而它的一些例行程序以及语

言编译程序等则存放在称为“系统带”的磁带上。监控程序还为操作员提供了一套控制命令。

监控程序的出现,明确了用户与操作员的分工,不仅提高了计算机系统的效率,而且方便了用户和操作员的使用。监控程序提供的文件系统能够按用户需要将程序和数据以文件形式保存起来,用户可以按文件名请求读写文件。但是,这样的系统还有很多缺点,例如:如果某一程序发生故障,系统就会停下来;它不允许用户干预自己程序的运行,用户没有“临场指挥权”;如果某一程序进入死循环,系统自己发现不了而“傻做”,而操作员因不了解程序逻辑同样在“傻等”;此外,它也无法防止错误程序对计算机系统的破坏。

监控程序在第二代计算机时期盛行一时。随着硬件技术的发展,计算机逐渐成为主机、内存、外存、通道、网络和各种配套设备的组合体,发展成硬件和软件的综合系统。面对发展变化了的环境,为了能够进行更加复杂的控制,人们不断增强监控程序功能。

### 3. 操作系统的诞生

在监控程序下运行一道程序,计算机的多数资源仍然不能充分利用。随着 CPU 速度的迅速提高,因等待外设而造成 CPU 不能被有效利用的问题日渐突出,于是形成了多道程序设计(即并行执行多个程序)的技术。这对计算机资源管理程序的发展产生深远的影响。

另一方面,计算机已经显露出来的能力促使人们产生利用它解决更大、更复杂问题的欲望,于是程序变得越来越大。尽管计算机的内存也在增大,但是有限的内存仍然满足不了运行大程序的需要。于是出现了虚拟存储技术:将程序分成若干部分,由计算机系统自动地完成程序各部分在内外存之间交换。

对多道程序设计和虚拟存储等计算机新技术的探索,最终导致了操作系统的诞生。1964年4月,IBM公司推出了IBM 360,标志着第三代计算机的诞生。IBM 360系统首次配上了功能强大的操作系统IBM OS/360,提高了整个计算机系统的效率。

操作系统的形成是一个渐进的过程。随着计算机应用范围的不断扩大,计算机运行速度的不断提高以及计算机系统功能的不断增强,系统可同时运行的程序由单道到多道,要求用户对机器的了解由多到少,管理和控制系统的软件由简单到复杂,终于“进化”成了操作系统。操作系统作为一种系统软件,为其他系统软件和应用软件的运行构筑了一个坚实而灵活的舞台,而且在使用中不断发展和完善着自身。

## 1.2.2 操作系统的基本功能

从资源管理的角度来考虑,操作系统包括处理器管理、存储器管理、文件管理、设备管理和作业管理这5个方面的基本功能。

### 1. 作业管理

用户请求计算机系统完成的计算任务称为作业,作业由程序和数据等组成。作业管理的一个重要功能是命令处理,即获取和分析来自键盘的命令或用户作业说明书中的作业控制命令,并做出相应的处理。作业管理的另一重要功能是作业调度,即根据预定的策

略选择适当的若干个作业进入内存运行,以使系统在资源使用率、系统吞吐量和用户满意度诸方面达到预期的要求。所谓吞吐量,是指在给定时间内所能完成的总作业量。

## 2. 处理器管理

处理器是计算机系统中最重要和最宝贵的硬件资源。为了提高处理器的利用率,提出了进程的概念,采用了多道程序等技术。操作系统为每个运行程序建立若干进程,这些进程间既有合作又有竞争。处理器管理的主要功能包括进程控制(负责进程的创建、撤销和状态转换)、进程同步(负责协调并发执行进程间产生的同步和互斥关系)、进程通信(负责合作进程间进度协调和信息传递)和进程调度(分配处理器给进程并保证进程顺利执行)。

## 3. 存储器管理

存储器管理的重点是内部存储器管理。系统中不断地完成已有作业,接收新作业,作业的内存要求各不相同,操作系统必须正确地分配和回收内存。通常,程序中所表示的地址不同于该程序运行时的实际内存地址,操作系统要负责这两种地址之间的转换。当多个程序共享内存时,操作系统要保证它们彼此隔离、互不干扰。当用户程序的存储需要量超过系统当前可提供的内存量时,操作系统必须设法利用外存来满足用户程序的需要。简而言之,存储器管理的主要功能包括存储分配、存储保护和存储扩充。

## 4. 文件管理

操作系统一般将需要长期保存的程序和数据组织成文件,存放在外部存储器上,使用户能够方便地按文件名进行存取,并保证文件的安全和共享。文件管理至少应包括文件空间的管理、目录管理、文件操作管理和文件保护等功能。

## 5. 设备管理

现代计算机系统一般采用中断和通道技术并通过控制器把各种外部设备和 CPU 连接起来。设备管理的任务就是使一般用户不必了解详细的接口就可方便地使用设备,并提高系统中各设备的利用率。在多用户系统中,设备管理的另一任务是保证多个用户能够共享设备。设备管理要合理地分配通道、控制器和 I/O 设备,并按用户的 I/O 请求执行具体的 I/O 操作。所以说,设备管理的主要功能包括设备分配、传输控制和设备独立性保证。

上述 5 个方面的功能,是就一般而言的。对于具体的操作系统,可能还有其他特殊要求。例如网络操作系统还要能够管理网络。但是,为特定目的和特定配置设计的操作系统不需要具备所有的功能;功能俱全的操作系统代码量必然极大,本身就需要占用大量的系统资源;而且,功能繁多的系统本身结构必然复杂,仅系统内部的管理就会需要很大的开销。因此,具体的操作系统应该根据不同目的在功能上有适当的取舍。

### 1.2.3 操作系统的基本特征

与其他软件相比较,操作系统具有如下基本特征。

#### 1. 并发性

在操作系统中,所谓并发性是指让多个程序同时在系统中运行。程序的并发性具体

体现在如下两个方面：用户程序与用户程序并发执行，用户程序与操作系统程序并发执行。在单 CPU 系统下，宏观上看，并发程序同时在向前推进，实际上是交替在 CPU 上运行。在多 CPU 系统中，并发程序不仅在宏观上是并发的，而且在微观上，即在处理器一级上也是并行运行的。而在分布式系统中，多台计算机的并存，使程序的并发特征得到更充分的体现。

这里所说的并发都是在单个操作系统统一指挥下的并发。在两个独立的操作系统控制下的计算机上，它们的程序也是并行运行的，但这并不是我们要讨论的情况。这里提到的并行性和并发性这两个概念是有区别的：并行性是两个或多个事件在同一时刻发生，这是一个具有微观意义的概念；而并发性是指两个或多个事件在同一时间间隔内发生，如果称之为同时，则是宏观意义上的同时。并行的若干个事件是并发的，而并发的若干事件则不一定是并行的。

## 2. 共享性

共享性是指操作系统程序与多个用户程序共用系统中的资源。这种共享是在操作系统控制下实现的。这种共享包括几方面的内容：操作系统要管理并发程序对处理器资源的共享，即在多个并发程序间通过调度来分配处理器时间；操作系统要负责管理对内存的共享和对外存的共享，并且保证对系统数据共享的正确性以及数据的完整性；操作系统要管理各种外部设备的共享。

共享可分为互斥共享和同时共享。系统中打印机、磁带机、扫描仪和重要系统数据等资源，虽然可以供多个用户程序共同使用，但是在一段特定时间内只能由一个用户程序使用。当这个资源正在被使用时，其他请求该资源的程序必须等待，只有等这次使用结束后才由操作系统根据一定的策略再选择一个用户程序占有该资源。通常把这样的共享称为互斥共享。系统中还有一类资源，它们在同一段时间内可以被多个程序同时访问。需要说明的一点是，这种同时是指宏观上的同时，微观上这些程序访问这个资源有可能还是交替进行的，而且它们交替访问的顺序不影响访问结果的正确性。硬盘是一个典型的可以同时共享的资源，那些可重入操作系统代码也是可以被同时共享的。

## 3. 虚拟性

操作系统向用户提供了远比直接使用裸机简单方便的高级抽象服务，隐藏了对硬件操作的复杂性，这就相当于把裸机变成功能更强而易于使用的虚拟机。操作系统本身又往往采用分层结构，其不同层次分别供不同类型的用户使用，因此，操作系统提供的是多层虚拟机。操作系统的虚拟性还体现在 CPU、内存、设备和文件管理等各个方面。例如，分时技术将计算机系统虚拟为多台逻辑上独立、功能相同的虚拟机，虚拟内存技术大大提高了内存使用效率，SPOOLing 系统将一台 I/O 设备虚拟为多台逻辑设备。此外，还有虚拟目录、虚拟文件等。

## 4. 不确定性

操作系统运行在一个随时变化的环境中，不可能对所运行程序的行为以及硬件设备的情况作任何假定。一个设备可能在任何时候向处理器发出中断请求，也无法肯定运行

着的程序会在什么时候做什么事情。因而,一般无法确切知道操作系统正处于何种状态。这就是不确定性的含义。但这并不是说操作系统无法很好地控制资源的使用和程序的运行,而是强调操作系统的设计与实现要充分考虑各种可能性,以便稳定、可靠、安全、高效地达到程序并发和资源共享的目的。

## 1.3 操作系统的结构与分类

对于人们根据不同环境和目的设计出来的各种操作系统,从不同的角度出发,就会得到不同的分类。例如,从使用环境和作业处理方式来考虑,操作系统可分为批处理操作系统、分时操作系统和实时操作系统;根据所支持的用户数目,可分为单用户操作系统和多用户操作系统;根据硬件结构,可分为网络操作系统、分布式操作系统、嵌入式操作系统和多媒体操作系统等。下面介绍的操作系统各种类型,不是按单一标准划分的结果。某个特定的操作系统可以归入若干类型之中。

### 1.3.1 操作系统的结构

软件结构是影响软件质量的内在因素,结构良好的软件可以提高可靠性、可维护性和工作效率。操作系统是由很多相关模块组成的庞大程序,它的每个模块的质量固然重要,但是,它的整体结构更是起着决定性的作用。

#### 1. 无序模块式

早期的操作系统大多采用无序模块式结构,现在有些小型操作系统仍采用这种结构。这些操作系统中,每个模块都有良好定义的接口,相互间的调用不受约束。这种方法的主要优点是结构紧密,组合方便,灵活性大;主要缺点是模块独立性差,结构不清晰,难以保证可靠性。

#### 2. 分层式

荷兰计算机科学家狄克斯特拉(E. F. Dijkstra)在1968年首先提出了操作系统层次结构法。他采用这种方法亲自设计了一个操作系统THE。THE操作系统是一个多道批处理操作系统,共有15个模块,其中10个是系统模块,5个是用户模块,所有这些模块构成THE的6个层次(如图1-3所示),系统严格遵循单向依赖的原则。即只允许上层模块调用下层模块,不能反向调用。虽然THE操作系统比较简单,但它所演示的方法对后来的操作系统结构设计产生了巨大的影响。

此后,汉森(Hansen)等人使用进程、类程和管程来构造操作系统。进程包括可执行程序、数据、程序计数器、栈指针、寄存器和程序运行需要的所有信息。类程是一种管理专用资源的模块。管程是一种特殊的模块或软件包,由过程、变量和其他数据结构等组成。1975年前后,汉森在PDP 11/45机上实现了3个小系统。其中,使用并发Pascal编译系统研制的单用户操作系统Solo只花了2人年。Solo系统分成7个管程,7个进程和10个类程,共24个模块,分7层(如图1-4所示)。

|       |             |
|-------|-------------|
| 第 5 层 | 用户模块        |
| 第 4 层 | 设备管理        |
| 第 3 层 | 控制台打字管理     |
| 第 2 层 | 存储管理        |
| 第 1 层 | CPU 调度和多道程序 |
| 第 0 层 | 硬件          |

图 1-3 THE 的层次结构

|       |       |
|-------|-------|
| 第 6 层 | 进程    |
| 第 5 层 | 缓冲区模块 |
| 第 4 层 | 程序管理  |
| 第 3 层 | 磁盘管理  |
| 第 2 层 | 控制台管理 |
| 第 1 层 | 资源管理  |
| 第 0 层 | 内核    |

图 1-4 Solo 的层次结构

被广泛接受的操作系统分层结构如图 1-5 所示,称为洋葱头式分层结构。“洋葱头”的中心是裸机本身提供的各种功能,向外扩展的每一层都提供一组功能,这组功能只依赖于该层以内的层次。“洋葱头”的各层组成了连续的一系列虚拟机器,紧挨着裸机的是操作系统内核,而最外层就是具备用户所需功能的虚拟机器。

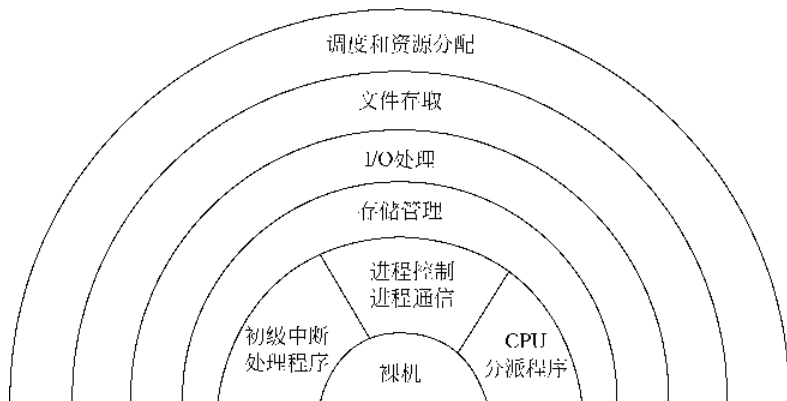


图 1-5 洋葱头式分层结构

### 3. 虚拟机式

如果换个角度看,前面谈到的分时系统具有两方面功能:一个是支持多道程序,另一个是向多个联机用户各提供一台方便的虚拟机器。在分时系统中,实现这两方面功能的模块合并为一个整体,因此系统非常复杂,维护很困难。而且,虽然各个用户都感觉在单独使用一台机器,但大家只能使用唯一的一种虚拟机器,缺乏灵活性。

虚拟机式操作系统的设计思想是将分时系统的上述两方面功能彻底隔离开来。这种操作系统的核心称为虚拟机监控程序,在裸机上运行,具有多道程序功能。它向上层提供了若干台虚拟机,每台虚拟机仅仅包含 CPU 工作状态、I/O 功能、中断和其他真实硬件所具有的功能,不具有文件等高级功能,称为会话监控系统。在不同的会话监控系统上可以运行不同的操作系统,形成更高一层的虚拟机(如图 1-6 所示)。支持多道程序和提供虚拟机器这两方面功能完全分开后,两者都较简单、较灵活和易于维护。

洋葱头式分层结构的操作系统由内向外形成一层层虚拟机器。但是,它们具有以下明显的区别。