

第 3 章 操作系统基础

3.1 操作系统概述

操作系统 (Operating System, OS) 对人们来说并不陌生, 实际上, 绝大多数用户在使用计算机时, 并不是直接与计算机沟通, 而是通过操作系统来使用计算机的。也可把操作系统形象地视为计算机的“总管”, 它负责管理其他“员工”, 并把主人的要求传递给他们。例如, 用户希望运行一个程序, 就把包含这个程序的文件名告诉操作系统, 再由操作系统去运行这个程序; 如果想编辑一个文件, 也要告诉操作系统文件名是什么, 它会启动相应的编辑器, 以便对那个文件进行处理。没有操作系统, 大多数用户根本无法使用计算机。

3.1.1 计算机系统的层次结构

通过前面章节的学习, 已知一个完整的计算机系统是由硬件系统和软件系统两大部分构成的, 那么这两部分在计算机中是如何组织在一起的呢? 可以认为是按照一定规则分层组织的, 图 3.1 描述了计算机系统的层次结构关系。

计算机系统可以划分为 4 个层次, 硬件、操作系统、系统实用软件和应用软件。最底层 (最内层) 是计算机硬件, 一台仅由硬件组成的计算机称为“裸机”。操作系统是紧贴在硬件之上的第一层软件, 是对硬件系统的第一次扩充。操作系统管理和控制下层的硬件系统, 同时向上层的系统实用软件和应用软件提供一个良好的使用环境。可以看出, 正是操作系统把一个裸机变成了操作方便、灵活的计算机系统。

例如, 在组装计算机时, 将所需要的硬件购买后组装在一起, 这时的计算机称为“裸机”, 不易使用。当安装操作系统后, 计算机的功能就增强了, 也称这种“新的功能更强的计算机系统”为“虚拟机”。系统实用软件由一组系统实用程序组成, 如语言处理程序、调试程序、DBMS 等。系统实用程序的功能是为应用软件提供服务, 支持其他软件的编制和维护。系统实用软件层位于操作系统之上, 它需要操作系统的支持。

在计算机系统层次结构中, 4 层间表现为单向服务关系, 即上层可以使用下层提供的服务, 但下层不能使用上层的服务。例如, 操作系统通过接口向上层软件 and 用户提供各种

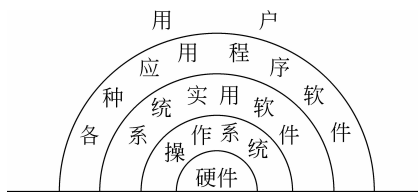


图 3.1 计算机系统的层次结构关系

服务,而上层软件 and 用户则通过操作系统提供的接口来访问硬件。

因此,操作系统在整个计算机系统中占据着非常重要的地位,它不仅是硬件与所有其他软件的接口,而且任何数字电子计算机都必须在其硬件平台上加载相应的操作系统之后,才能构成一个可以协调运转的计算机系统。只有在操作系统的指挥控制下,其他系统实用软件和应用软件才能获得运行条件。没有操作系统,任何软件都无法运行。

3.1.2 操作系统的定义

什么是操作系统?下面给出一种操作系统的定义。

操作系统是指控制和管理计算机系统硬件和软件资源,合理组织计算机的工作流程,方便用户使用计算机的一组程序和数据的集合,是计算机系统中最基本的系统软件。

可以从不同的角度来观察操作系统的作用。从资源管理的观点来看,操作系统是计算机系统资源的管理者,它好比计算机系统的“管家”,管理着计算机中的各种资源。从用户使用计算机的角度来看,操作系统是用户和计算机硬件系统之间的接口,是人机交互的桥梁,它负责向用户提供良好的使用界面。因此,操作系统的作用主要有如下两方面。

1. 管理系统中的各种硬件和软件资源

计算机系统中通常包含各种各样的硬件资源和软件资源,所有的硬件称为硬件资源,所有的程序和数据信息称为软件资源。使用计算机就是使用硬件资源和软件资源。操作系统是计算机系统资源的管理者和仲裁者,负责为运行的程序分配资源,对系统中的资源进行有效管理,使系统资源很好地为用户服务,保证系统中的资源得以有效地利用,使整个计算机系统能高效地运行。

2. 方便用户使用计算机

大多数用户都是通过操作系统来使用计算机的,因此一个好的操作系统应为用户提供良好的操作界面,使用户能够便捷、安全、可靠地操纵计算机硬件和运行自己的程序,而不必了解硬件和系统软件的细节就可方便地使用计算机。例如,计算机安装好 Windows 操作系统后,用户就可以在图形界面下用鼠标进行各种操作,而不用考虑计算机的硬件特性。

3.1.3 操作系统的引导

要想使用计算机,首先必须把操作系统调入内存,这个过程称为引导系统。在计算机电源关闭的情况下,打开电源开关启动计算机被称为冷启动;在电源打开的情况下,重新启动计算机被称为热启动。

每当启动计算机时,操作系统的核心程序及其他需要经常使用的指令就从硬盘调入内存。操作系统的核心部分功能就是管理 CPU、存储器和其他设备,维持计算机的时钟,调配计算机的设备、程序、数据和信息等资源。操作系统的核心部分是常驻内存的,而其他部分不常驻内存,通常存放在硬盘上,当需要的时候才调入内存。

无论计算机规模如何,其引导过程都是相似的。例如,一台装有 Windows 操作系统

的计算机冷启动时的过程如下。

- ① 计算机加电时,电源给主板及其他设备发出电信号。
- ② 电脉冲使处理器芯片复位,并查找含有 BIOS 的 ROM 芯片。BIOS 代表基本输入输出系统,是一段含有计算机启动指令的系统程序,它存放在一个 ROM 芯片中,所以也称为 ROM-BIOS。
- ③ BIOS 进行加电自检,即检测各种系统部件,如总线、系统时钟、扩展卡、RAM 芯片、键盘及驱动器等,以确保硬件连接合理及操作正确。自检的同时显示器会显示检测得到的系统信息。
- ④ 系统自动将自检结果与主板上的 CMOS 芯片中的数据进行比较。CMOS 芯片是一种特殊的只读存储器,其中存储了计算机的配置信息,包括内存容量、键盘及显示器的类型、软盘和硬盘的容量及类型,以及当前日期和时间等,自检时还要检测所有连接到计算机的新设备。如果发现了问题,计算机可能会发出“嘟嘟”声,显示器会显示出错信息,若问题严重,计算机还可能停止操作。
- ⑤ 如果加电自检成功,BIOS 就会到外存中去查找一些专门的系统文件(也称为引导程序),一旦找到,这些系统文件就被调入内存并执行。接下来,由这些系统文件把操作系统的核心部分导入内存,然后操作系统就接管、控制计算机,并把操作系统的其他部分调入计算机。
- ⑥ 操作系统把系统配置信息从注册表调入内存。在 Windows 中,注册表由几个包含系统配置信息的文件组成。在计算机的操作过程中,经常需要访问注册表以存取信息。例如已安装的硬件和软件、个人用户的口令、对鼠标速度的选取等信息。

当上述步骤完成后,显示器屏幕上就会出现 Windows 的桌面和图标。接着操作系统自动执行“启动文件夹”中的程序。至此,计算机就启动好了,用户可以开始操作计算机了。

3.1.4 操作系统的形成过程与特征

1. 操作系统的形成过程

操作系统从无到有,规模从小到大,功能从弱到强,形成过程大致经历了手工操作(第一代计算机时代,那时未产生操作系统)、管理程序(第二代计算机时代)和操作系统三个阶段。从第三代计算机开始,随着多道批处理、分时、实时操作系统的相继出现,标志着操作系统的正式形成。

2. 操作系统的特征

操作系统作为一类系统软件,其基本特征是并发性、共享性和不确定性。

① 并发性

指两个或多个活动在同一时间间隔中进行,对用户来说,好像同时都在使用计算机,这是宏观上的概念。而从微观上看,是多个进程交替使用 CPU。

② 共享性

指内存中的多个并发执行的程序共享计算机系统的各种资源,即系统中的资源被多

个任务所共用。可以共享的资源包括内存、CPU、外存等。

③ 不确定性

在多道程序环境下,系统中每个用户程序的运行时间、要使用的系统资源、资源使用多长时间等,操作系统在程序运行前是不知道的。有可能先进入内存的作业后完成,而后进入内存的作业先完成,也就是说,程序是以异步方式运行的,存在不确定性。

3.1.5 操作系统的分类

以功能的角度进行分类是被广泛采用的典型操作系统分类法,它把操作系统分为批处理操作系统、分时操作系统和实时操作系统等。

1. 批处理操作系统

批处理操作系统是最早问世的操作系统,分为单道批处理操作系统和多道批处理操作系统。早期,计算机只能通过控制台使用,而启动计算机软硬件均需要大量的启动时间。为了减少启动时间,计算机就需要由操作员来操作。用户把要计算的问题、数据和作业说明书一起交给系统操作员,由系统操作员将相同的一批作业输入计算机中,然后由操作系统调度各作业运行。

首先出现的是单道批处理操作系统,该系统(批处理系统既是操作系统的一种类型,又是对配置了批处理操作系统计算机的一种叫法)以成批的方式接收用户提交给计算机运行的作业,但系统每次只调一个用户程序进入内存让它运行,直到整个程序运行结束,或因为某种错误使用该用户程序无法继续运行而终止,然后计算机自动将同批作业的下一个作业调入内存运行。这样计算机系统不再需要等待人工操作,节省了作业之间的过渡时间,提高了计算机的整机利用率。

但是,单道批处理系统每次只运行一个作业,当运行中的作业进行输入输出操作时,处理机将处于空闲等待状态,这将浪费宝贵的处理机资源。于是就出现了多道批处理操作系统。

多道批处理操作系统保持了单道批处理操作系统中作业自动过渡的功能。此外,为了提高系统效率,还支持在内存中同时放入多道用户作业,并将各个作业分别存放在内存的不同部分,而这些作业可以获准占用处理机和外部设备。从微观上看,内存中的多道程序轮流地或分时地占用处理机,交替执行。每当运行中的一个作业因输入或输出操作需要调用外部设备,而使处理机出现空闲时,系统就自动进行切换,把处理机交给另一个等待运行的作业,从而将主机与外部设备的工作方式由串行改为并行,使处理机在等待外部设备完成任务时可以运行其他程序,从而显著提高了计算机系统的吞吐量,提高了系统资源的利用率。

但是,批处理操作系统有一个很大的缺陷,即在程序运行过程中不允许用户与计算机进行交互,程序或数据出现任何错误都必须待整个批处理结束之后才能修改,因此它不适宜处理在运行过程中需要用户加以干预的程序。但是,用户却希望能有一种方法,支持在程序运行过程中用户与计算机直接交互,这就导致了分时操作系统的出现。

2. 分时操作系统

所谓“分时”是指若干并发程序对 CPU 时间的共享。分时操作系统往往用于连接有多台终端的计算机系统,它允许多个用户(从一个到几百个)通过各自的终端同时交互地使用一个计算机系统。用户在各自的终端上输入命令、程序或数据,并以交互方式控制程序的执行。

在分时系统中,若干个终端用户作业被驻留在内存中,由系统根据某种策略(如优先权等)调度分配处理机资源,对作业进行处理。系统将处理机的工作时间划分为许多很短的时间片(就是一小段时间,如 10ms),每个用户作业占用一个时间片,各用户作业按一定顺序轮流占用处理机。换句话说,处理机轮流接收和处理各个用户从终端输入的命令,即按某个轮流次序在用户之间分配允许使用的处理机时间。

例如,一个带有 10 个终端的分时系统,若给每个用户每次分配 100ms 的时间片,则每隔 1s 即可为所有用户服务一遍。如果用户的某个处理要求时间较长,分配给它的一个时间片不足以完成该处理任务,则它只能暂停下来,等到下一个时间片轮到时再执行。

由于计算机运行速度极高,与用户的输入输出时间相比,轮换的时间间隔是极短的,所以系统每次都能对用户程序做出及时的响应,从而使每个用户都感觉似乎自己独占了整个计算机系统,好像在独享 CPU 资源。

分时系统提高了系统资源的共享程度,适用于程序调试、软件开发等需要频繁进行人机交互的作业,其典型代表是 UNIX 操作系统。

3. 实时操作系统

所谓“实时”,就是“立即”或“及时”,是指系统能够及时响应随机发生的外部事件,并以足够快的速度完成对事件的处理。随着计算机应用领域的日益广泛,实时操作系统应运而生,它具有时间性强、响应快、高可靠性等特点,常配置在需要“实时响应”的计算机系统上。

根据应用领域不同,可将实时系统分为实时信息处理系统和实时过程控制系统。

(1) 实时信息处理系统通常配有大型文件系统或数据库,事先存有经过合理组织的大量数据,能及时响应来自终端的服务请求,进行信息的检索、存储、修改、更新、加工、删除、传送等操作,并在很短的时间内对用户请求做出正确回答。其应用领域如情报检索、机票预定、银行业务、电话交换等。

(2) 实时过程控制系统又可分为两类:

① 以计算机为控制中枢的生产过程自动化系统,如机械加工、发电、冶炼、化工、炼油等自动控制。在这类系统中要求计算机及时采集和处理现场信息,进而控制相关的执行机构,使得某些参数,如流量、压力、温度、液位等按一定规律变化或保持不变,从而达到提高质量、增加产量及实现生产过程自动化之目的。

② 对飞行物体的自动控制,如对飞机、导弹、人造卫星的制导等。

4. 个人计算机操作系统

近年来由于 PC 的普及,人们对个人计算机操作系统更加熟悉,如 Windows 等。

(1) 单用户单任务操作系统

单用户单任务操作系统在某一时刻只允许一个用户运行一个程序,该用户独占计算机系统的全部硬件、软件资源。单用户单任务操作系统是为简单的小型计算机、微型计算机开发的。最早的操作系统就是单用户单任务操作系统,其特征是:个人使用、界面友好、管理方便,适于普及。例如曾在 PC 上广泛使用的 MS-DOS 就是单用户单任务操作系统。

(2) 多用户操作系统

多用户操作系统支持多个用户程序同时在系统中运行。计算机网络、中等规模的服务器、大型主机及超级计算机都允许成百上千个用户同时使用计算机系统,因此被称为多用户系统。多用户系统中的每个用户通过各自的终端运行自己的程序,操作系统负责分配资源和管理调度,使各用户程序互不干扰地运行。分时操作系统和网络操作系统都属于多用户操作系统,著名的 UNIX、XENIX 操作系统都是分时多用户操作系统。多用户系统除了具有界面友好、管理方便和适于普及等特征外,还具有多用户使用、可移植性好、功能强、通信能力强等优点。

(3) 多任务操作系统

多任务操作系统允许一个用户同时执行多个任务,即允许用户同时运行两个以上的应用程序。例如,用户可以一边收听计算机播放的音乐一边用文字处理软件进行 Word 写作,多媒体播放程序和文字处理软件 Word 可以并行地运行。

当同时运行多个程序的时候,只有一个程序在前台运行,其他程序均在后台运行。前台含有当前正在使用的程序,其他那些正在运行但非正在使用的程序处于后台。通过单击任务栏上的某个程序名可以方便地将其置于前台,而其他所有程序就都被置于后台。现在的大部分操作系统都是多任务的,如 OS/2、Windows XP、UNIX、Linux 等都是典型的多任务操作系统。

多任务系统基于对处理机的分时使用。前面所述的多道批处理系统和分时系统同时也是多任务系统。

5. 网络操作系统

随着计算机网络的发展,网络管理功能融入了操作系统,于是又出现了网络操作系统。网络操作系统与传统的单机操作系统有所不同,它是建立在单机操作系统之上的一个开放式软件系统,除了具有传统的单机操作系统的 5 大功能外,还具有网络管理功能,即:实现网络通信、网络资源共享和保护,提供网络服务和网络接口,以及方便用户使用网络中的各种软、硬件资源等。

目前,大部分操作系统如 UNIX、Linux、Windows 和 NetWare 等都具有网络管理和操作功能,可以用作网络操作系统。

6. 分布式操作系统

分布式系统是指以计算机网络为基础、能有效实现地域分布很广的若干计算机系统间的资源共享、并行工作、信息传输和数据保护等功能的一类系统构成,由此产生了这类系统的工作平台——分布式操作系统,该系统具有以下特征。

① 分布式处理

分布式操作系统可将所有系统任务提交给系统中的任何处理机运行,即将资源、功能、任务及控制等分散在各个处理单元上,自动实现全系统范围内的任务分配并自动调度各处理机的工作负载。对用户而言,实际上并不知道自己的程序是在哪台计算机上运行的,也不知道自己的文件存放在什么地方。

② 实施整体控制

分布式操作系统负责对各个分布的资源进行统一的整体控制,如调度系统中若干台计算机相互协作完成一个共同的任务。它与网络操作系统相比更着重于任务的分布性,即把一个大任务分为若干个子任务,分派到不同的处理站点上并行执行,充分利用各计算机的优势。在分布式操作系统控制下,使系统中的各台计算机组成了一个完整的、功能强大的计算机系统。所以,在用户看来分布式系统如同传统的单 CPU 系统,实际上它由众多处理器组成,每一个处理机上都运行该操作系统的一个副本。

7. 嵌入式操作系统

嵌入式操作系统(Embedded Operating System, EOS)负责其所在系统的全部软、硬件资源的分配与调度工作,控制、协调并发活动,同时体现系统的特征,并能通过装卸某些模块达到系统所要求的功能。EOS 在系统的实时高效性、硬件的相关依赖性、软件固态化以及应用的专用性等方面具有较为突出的特点。

EOS 过去主要应用于工业控制和国防系统领域,随着 Internet 技术的发展以及 EOS 的微型化和专业化,EOS 开始从单一的弱功能向高专业化的强功能方向发展,其应用平台之一是各种家用电器和通信设备,例如手机、遥感控制系统等。由于家用电器市场比传统的计算机市场大很多,因此嵌入式操作系统逐渐成为操作系统发展的另一个热门方向。

综上所述,一个具体的计算机操作系统,它可能不纯属于其中某一类型,而是同时具备其中几类的特点,也可能以某一类为主,兼有其他类的特点,这样就形成了通用操作系统。这是由于不同类型操作系统间的差异,主要是系统侧重面的不同造成的。如批处理系统重点放在计算机资源的利用率和作业的吞吐能力上,分时系统重点放在交互作业和响应时间上,而实时系统则侧重于系统的完整性、可靠性、严格的响应时间等问题。

3.2 操作系统的功能

通过前面的学习已知,没有任何软件的计算机叫“裸机”,在裸机上配置了操作系统后就构成了操作系统虚拟机,而用户就在这个扩充后的计算机上工作。操作系统为用户提供了一个基础性的工作平台,了解操作系统的一些基本功能和概念,将有助于更深刻地理解计算机和更主动地使用计算机。

操作系统有许多不同的种类,但大多数操作系统都具有以下 5 个方面的管理功能,即进程与处理机管理、存储器管理、设备管理、文件管理和用户接口,它们与硬件的对应关系如图 3.2 所示。

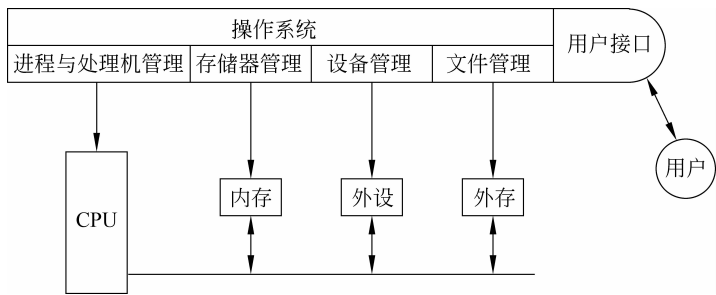


图 3.2 操作系统管理功能与硬件对应关系

3.2.1 进程与处理机管理

1. 进程

进程(process)是操作系统中的一个核心概念,其定义为,进程是程序的一次执行过程,是系统进行调度和资源分配的一个独立单位。

操作系统对每一个执行的程序都会创建一个进程,一个进程就代表一个正在执行的程序。操作系统正是采用进程概念来描述一个程序的执行过程、记录执行程序的相关信息,并实现对一个正在执行的程序的管理。

为了更好地理解进程的概念,有必要对操作系统中“进程”和“程序”这两个概念做一个对比。

“程序”是一种静态的概念,是指存储在文件中的程序,如源程序、可执行程序等。对于程序文件,可以进行创建、编辑、编译、复制、删除等操作。

而“进程”是动态的概念。当用户运行一个程序时,系统就为它建立一个进程,并为该进程分配内存、CPU 和其他资源。当程序运行结束时,为该程序本次执行所建立的进程就消亡了,进程有它自己的生命周期。对于进程(或者说对于一个正在执行的程序),有另外一套不同于程序文件的操作,例如,可以观察进程信息(查看当前有哪些程序正在执行),可以撤销一个进程(中止一个程序的执行),可以挂起一个进程(暂时停止一个程序的执行),可以在进程之间进行切换(如在字处理软件和电子表格软件之间来回操作)。

由于用户使用计算机的主要方式就是执行程序,了解进程的基本概念将有助于更加清晰地了解程序执行过程中所发生的一些事情,有助于更好地操纵和控制一个正在执行的程序,有助于更加有效地使用针对进程的系统命令和工具。

2. 进程控制块(PCB)

系统如何建立一个进程呢? 当启动一个程序时,系统就为之建立一个进程控制块(process control block,PCB),这就是进程存在于系统中的实体。

PCB 相当于一张电子表格,它记录了该进程的描述信息、控制信息和资源信息,其中一些主要内容如下:

- 进程标识名(系统中唯一标识一个进程的名字)。
- 进程所属的用户(用户名)。

- 进程当前状态(系统根据进程状态决定如何控制进程)。
- 进程优先级(进程调度的依据)。
- 进程的起始地址(该进程代码从内存的哪个地址开始执行)。
- 进程使用资源的信息(使用了哪些外部设备、文件等)。
- CPU 现场保护信息(当进程暂停运行时,要保留 CPU 现场,如执行到哪条指令、当前有关寄存器中的内容等,以便该进程能从断点处再次恢复执行)。

PCB 是系统感知进程存在的唯一实体。操作系统通过访问 PCB,就知道有哪些进程存在、每一进程当前处于什么状态,就可以对进程进行调度、为进程分配软硬件资源。当进程结束时,系统通过释放 PCB 来回收进程所占用的各种资源。

3. 进程状态

在进程管理中,进程状态是一个核心概念,下面结合图 3.3 简要介绍进程的几种主要状态及其变化因素。

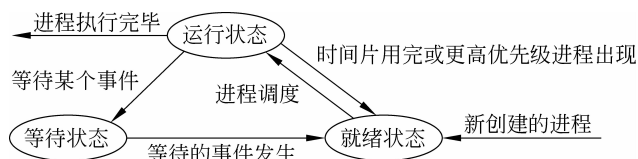


图 3.3 进程调度与进程转换

由于在系统中可以同时存在多个进程,而进程之间又存在着相互制约关系(如对资源的争夺),所以一个进程并不是自始至终都在运行,而是“运行—暂停—运行”,即“停停走走、断断续续”的,进程在它的生命周期中所处的状态是不断变化的,这也就是进程的动态性。下面是进程 3 种基本状态的含义。

(1) 就绪状态

处于该状态的进程除 CPU 外,其他所需资源已全部满足,换句话说,处于就绪状态的进程只缺 CPU 资源,一旦获得 CPU,进程就可以转换为运行状态,进程对应的代码被执行。当用户启动一个程序时,系统为它创建一个进程,并将其置为就绪状态。

(2) 运行状态

正在 CPU 上运行的进程所处的状态就是运行状态。当一个进程处于运行状态时,CPU 就会执行该进程的指令代码(取指并执行)。当计算机只有一个 CPU 时,同一时刻最多有一个进程处于运行状态,而在多 CPU 系统中,可以有多个进程同时处于运行状态(每个 CPU 上都运行着一道程序)。

(3) 等待状态

当一个进程由于等待某个事件而不能运行时,即处于等待状态(又称阻塞状态或挂起状态)。处于等待状态的进程即使给它 CPU 也不能运行,因为缺少其他必要条件。在这里,“事件”可以是某个外部设备操作的完成,也可以是系统某项工作的完成,还可以是有协作关系的其他进程的工作等。

4. 进程转换与操作系统调度策略

进程与处理机管理的核心工作就是进程调度,即按照一定的法则将 CPU 动态地分配给某个就绪进程。下面结合进程状态转换介绍进程调度的几种策略。

(1) 就绪状态→运行状态

在进程调度算法中,操作系统最常采用的一种算法就是优先级调度法。如果处在就绪状态的进程不只一个,它们就按照进程优先级排队等待 CPU,优先级高的进程排在前面。进程的优先级是在创建进程时由系统指定的。一旦 CPU 出现空闲,系统就将 CPU 分配给排在前面的进程,而获得 CPU 的进程也就从就绪状态变为运行状态。

(2) 运行状态→等待状态

处于运行状态的进程在代码执行过程中可能会发出对某种资源的请求,如访问某个外部设备、访问某个文件或与其他进程进行数据交换等。进程在发出这些请求后,必须要等待所需资源得到满足、相关操作完成之后,代码才能继续执行下去。一条指令执行过程中要等待的条件是多种多样的,按照操作系统的术语,把这些等待因素归结为等待某个事件的发生。例如,程序代码执行到一条输入指令,要从键盘(外部设备)读入一个数据,程序就要等到用户从键盘输入数据并接收到数据后才算完成一条输入指令,程序也才能往下执行。

程序执行一条 I/O 指令所需的时间远比执行一条算术逻辑指令的时间要长,很多外部设备都是机电结构的,它们的工作速度与 CPU 的工作速度相比不在一个数量级上。因此,让一个进程在占据 CPU 资源的情况下等待一条输入输出指令的完成是不合算的,如果将执行一条 I/O 指令时等待外部设备完成相关操作的时间用来执行其他进程的代码,显然可以执行的指令数量是相当可观的。

因此,处于运行状态的进程在执行到一条 I/O 指令或其他需要等待某一条件的指令时,该进程从运行状态转为等待状态,而把 CPU 释放出来让给其他进程。这样的调度法则就可以让 CPU 尽量不要空闲,以充分发挥 CPU 的功能。

进程由运行状态转为等待状态是由于进程自身的操作所引起的,是因为进程执行到某条指令时,由于要等待某个事件而不得不停下来,而该条指令实际上也是处于执行一半的状态。所以当进程由运行状态变为等待状态时,由于 CPU 要让给别的进程使用,当前现场信息必须完整地保护好(如当前中断的是哪一条指令?当前各寄存器的内容是什么?),以便将来恢复现场,使被中断的进程得以继续执行。

(3) 运行状态→结束

处于运行状态的进程如果在执行期间刚好完成任务(代码执行完),该进程的生命周期也就结束了,系统将回收该进程占用的各种资源,删除进程控制块,该进程在系统中也就不复存在了。

(4) 等待状态→就绪状态

处于等待状态的进程在等待的事件发生后,就又具备了继续执行的条件,该进程的状态也就由等待状态变为就绪状态。例如进程执行到一条向打印机输出的指令后处于等待状态,当打印机回答输出操作已经完成,该进程等待的事件就发生了。

处于等待状态的进程可能不只一个,每个进程都在等待自己盼望的事件发生。哪个