

第5章 操作系统

操作系统是全面管理计算机软件 and 硬件资源的系统程序,能够合理地组织计算机的工作流程,使计算机的各部件互相协调一致地工作。操作系统也是用户与计算机之间的接口,通过它用户可以更方便、更有效地使用计算机资源。

半个多世纪以来,从最早的单道批处理系统、分时操作系统,到今天的微机操作系统,面向大型机的多任务、多用户操作系统,网络操作系统、分布式操作系统及嵌入式操作系统,它在调度和控制计算活动,提供软件开发、运行和应用环境,挖掘计算机潜力,提高计算机的性能等方面发挥着越来越重要的作用。

本章主要介绍操作系统的基本概念、功能以及基本原理与方法,包括进程管理、存储管理、文件管理和设备管理等。最后介绍 CP/M、DOS、Windows、UNIX、Linux 等典型操作系统的发展历程。

5.1 操作系统概论

5.1.1 操作系统的功能

从一般用户角度看,可把操作系统看作用户与计算机系统的接口。操作系统为用户提供以下几个方面的服务。

(1) 程序开发。操作系统提供各种工具和服务,如编辑器和调试器,用于帮助程序员开发程序。

(2) 程序运行。运行一个程序需要很多步骤,必须把指令和数据载入到主存储器、初始化 I/O 设备、准备一些其他资源。操作系统为用户处理这些问题。

(3) I/O 设备访问。每个 I/O 设备的操作都需要特有的指令集或控制信号,操作系统隐藏了这些细节,并提供统一的接口。

(4) 文件访问控制。向用户提供方便的使用文件的接口,并采取一定的安全措施实现文件的共享与保护。

(5) 系统访问。对于共享和公共系统,操作系统控制对整个系统的访问,提供对资源和数据的保护,避免未授权用户的访问,还必须解决资源竞争时的冲突问题。

(6) 错误检测和响应。计算机系统运行时可能发生各种各样的硬件和软件错误,如存储器错误、设备故障、算术溢出、访问被禁止的存储单元等。对每种情况,操作系统都必须提供响应,使其对正在运行的应用程序影响最小。

从资源管理角度看,则可把操作系统视为计算机系统资源的管理者。计算机系统通常包含各种各样的硬件和软件资源,操作系统的任务就是在相互竞争的进程之间有序地

管理和分配处理器、存储器以及 I/O 设备等资源。资源管理主要包括以下工作。

(1) 跟踪记录资源的使用情况。在多任务系统中,系统资源要满足多个任务的要求,这就需要对资源使用情况进行跟踪和记录,了解当前资源状态或剩余情况,以满足任务对资源的请求。

(2) 分配和回收资源。在条件满足的情况下,将资源分配给请求的任务;根据任务完成的情况适时地回收系统资源,保证新任务的请求。

(3) 提高资源的利用率。在操作系统的管理下使系统资源得到合理、高效的使用。

(4) 协调多个任务对资源请求的冲突。当少量资源被多个任务请求时,就会产生冲突,这时操作系统需要分析请求任务的特性,对资源使用做出决策,协调各任务合理地使用资源。

5.1.2 操作系统的特征

操作系统的特征主要体现在并发性、共享性、虚拟性和异步性 4 个方面。

1. 并发性

并发性是指两个或两个以上的事件在同一时间段内发生。并发性体现了操作系统同时处理多个活动事件的能力。对只有一个处理器的系统,在一个时间段内,可以同时运行多个进程,实现多进程并发。这些并发进程体现:宏观上同时执行,微观上任何时刻只有一个在执行。通过并发,能够减少计算机中各部件间由于相互等待而造成的资源浪费,改善资源利用率,提高系统的吞吐量。并发性的实现比较复杂,需要解决进程之间的运行切换、进程内容保护、相互依赖进程之间的同步关系、进程资源分配的协调等问题。

2. 共享性

共享性是指计算机系统资源能够被并发执行的多个进程共同使用。操作系统对这些资源进行合理的调配和管理,并使得并发执行的多个进程能够合理地共享这些资源,达到节约资源、提高系统效率的目的。实现资源共享需要解决的问题有资源分配优化、信息保护、存取控制、进程之间同步等。因为进程的并发才会出现资源的共享,而只有资源有效的共享,才能保证并发的顺利执行。

3. 虚拟性

虚拟性是指操作系统通过某种技术将一个实际存在的实体变成多个逻辑上的对应体。这样的多个逻辑对应体可以被多个并发进程访问,提高了实体的利用率。虚拟性是操作系统管理资源的一种重要手段,其目的是为用户提供方便高效的资源利用。在计算机系统中,处理器、存储器、I/O 设备以及窗口、用户终端等都可以通过虚拟技术提供给并发的进程使用。与虚拟性相关的技术问题有处理器管理、虚拟存储器管理、SPOOL (simultaneous peripheral operation online) 技术等。

4. 异步性

异步性也称为随机性,是指在多道程序环境中多个进程的执行、推进和完成时间都是随机的、交替的、不可预测的。多个并发的进程由于受到资源限制不能一贯到底,只能“走走停停”。这种异步方式的进程执行,导致的后果是进程执行的最终结果不可重现。异步性会给操作系统带来潜在的危险,有可能导致并发程序的执行产生与时间有关的错误,所以操作系统必须采取一定的措施,保证在运行环境相同的情况下,多次运行同一程序,都会获得完全相同的计算结果。

5.2 操作系统的发展与分类

5.2.1 手工处理阶段

20世纪40—50年代,是电子管计算机时代,计算机运算速度慢(只有几千次/秒)。在那个年代里,同一组人设计、建造、编程、操作并维护一台机器。所有的程序设计是用纯粹的机器语言完成的,常常连线到插件板上以便控制机器的基本功能。那时没有程序设计语言,也没有操作系统。

到了20世纪50年代早期出现了穿孔纸带(或卡片),这时就可以将程序写在纸带(或卡片)上,然后读入计算机,不必再用插件板。用户先把程序纸带(或卡片)装上计算机,然后启动输入机把程序和数据送入计算机,接着通过控制开关启动程序运行。计算完毕,打印机输出计算结果,用户取走并卸下纸带(或卡片)。这种由单道程序独占计算机及人工操作的情况,在计算机速度较慢时是允许的,因为此时计算机完成一个任务所需要的时间相对较长,手工操作时间所占的比例还不是很大。

5.2.2 批处理系统

20世纪50年代中期,出现了晶体管计算机,计算机变得更可靠,运行速度也有了很大提高,从每秒几千次发展到每秒几十万次、上百万次。这时,手工操作已无法满足计算机高速度的需要,促使人们去开发一种软件来管理软硬件资源,以提高主机的使用效率。这样就出现了批处理系统(batch processing system)。

批处理系统克服了手工操作的缺点,实现了作业的自动过渡,改善了主机CPU和输入输出设备的使用情况,提高了计算机系统的处理能力。批处理系统的主要思想是使用一个称为监控程序的软件,它的内存分布如图5-1所示。通过使用这种操作系统,用户不再直接访问计算机,而是把卡片或磁带中的作业提交给计算机操作员,由他把这些作业按顺序组织成一批,并将整个作业放在输入设备上,供监控程序使用。每个程序完成处理后返回到监控程序;同时,监控程序自动加载下一个程序。

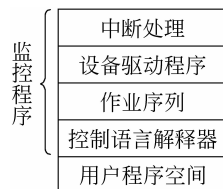


图 5-1 简单批处理系统的内存分布

早期的批处理分为两种方式：联机批处理和脱机批处理。

1. 联机批处理

输入输出设备和主机直接相连。作业的执行过程如下。

- (1) 用户提交作业,包括程序、数据以及用作业控制语言编写的作业说明书。
- (2) 作业被输入穿孔纸带或卡片。
- (3) 操作员有选择地把若干作业合成一批,通过输入输出设备(纸带输入机或读卡机)把它们存入磁带。
- (4) 监控程序读入一个作业。
- (5) 从磁带调入汇编程序或编译程序,将用户作业源程序翻译成目标代码。
- (6) 连接装配程序,把编译后的目标代码及所需要的子程序装配成一个可执行的程序。
- (7) 启动执行。
- (8) 执行完毕,由善后处理程序输出计算结果。
- (9) 再读入一个作业,重复第(5)~(9)步。
- (10) 一批作业完成,返回到第(3)步,处理下一批作业。

这种联机批处理方式解决了作业自动转接的问题,从而减少作业建立和人工操作时间。但是在作业输入和执行结果的输出过程中,主机 CPU 仍处在停止等待状态,这样慢速的输入输出设备和快速的主机之间仍处于串行工作模式,CPU 的时间仍有很大的浪费。由于系统对作业的处理都是成批地进行,且在内存中始终只保持一道作业,因此也被称为单道批处理系统。

2. 脱机批处理

脱机批处理方式的显著特点是增加一台不与主机直接相连而专门用于与输入输出设备打交道的卫星机,早期脱机批处理模型如图 5-2 所示。

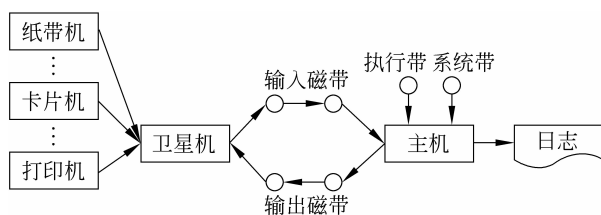


图 5-2 早期脱机批处理模型

卫星机的功能如下。

- (1) 输入设备通过它把作业输入到输入磁带。
- (2) 输出磁带通过它将执行结果输出到输出设备。

这样,主机不直接与慢速的输入输出设备打交道,而是与速度相对较快的磁带机发生关系。主机与卫星机可以并行工作,因此早期脱机批处理与早期联机批处理相比大大提高了系统的处理能力。

20 世纪 50 年代末至 20 世纪 60 年代初,出现了许多成功的批处理系统。第一个批

处理系统产生于20世纪50年代中期,由通用汽车(General Motors)公司开发,用于IBM 701计算机上;这个系统随后经过进一步改进,被IBM用户应用在IBM 704上;20世纪60年代初期,许多厂商自行开发了批处理系统,比较著名的是IBM公司开发的用于IBM 7090/7094机上的操作系统IBSYS,对其他系统有着广泛的影响。

脱机批处理也存在着一些缺点。首先是磁带需要人工拆卸,极其不便;其次是系统保护问题越来越突出。

20世纪60年代初期,硬件技术的发展出现了通道和中断技术,这两项重大成果导致操作系统进入执行系统阶段。通道是一种专用的处理部件,它能控制一台或多台外部设备的工作,负责外部设备和主机之间的信息传输。它受CPU控制,一旦被启动就能独立于CPU运行,这样CPU和通道就可以并行操作,而且CPU和各种外部设备也能并行操作。中断是指当主机接到某种信号(如I/O设备完成信号时)马上停止原来的工作转去处理这一事件,当事件处理完毕,主机又回到原来的工作点继续工作。

在通道与中断技术的基础上,I/O工作可以在主机控制之下完成,这与早期联机批处理系统相比有本质的区别。因为在通道与中断技术的支持下,不仅实现了联机控制输入输出工作,而且还实现了CPU和I/O的并行操作。执行系统可以这样来描述:借助通道与中断技术由主机控制I/O传输,监督程序不仅负责作业的自动调度,还要负责I/O控制功能。这个优化后的监督程序常驻内存,称为执行系统。

执行系统节省了卫星机,降低了成本,实现了主机和通道、主机和外部设备的并行操作,提高了系统的安全性。同时负责用户的I/O传输工作,检查用户I/O命令的合法性,避免了由于不合法的I/O命令造成的对系统的威胁。

执行系统的普及实现了标准文件管理系统和外部设备的自动调节控制功能。在这期间程序库变得更加复杂和庞大。随机访问设备(如磁盘磁鼓)已开始代替磁带作为辅助存储器;高级语言也比较成熟和多样化。20世纪50代末到20世纪60年代初期,出现了许多成功的批处理操作系统,其中IBM 7090/7094计算机配置的IBM OS是最有影响的。

5.2.3 多道程序系统

在单道批处理系统中,内存中仅有一道作业,这使系统中仍然有较多的空闲资源,致使系统的性能较差。为了进一步提高资源的利用率和系统的吞吐量(系统在单位时间内所完成的总工作量),在20世纪60年代中期引入了多道程序设计技术,因此形成了多道程序系统。在多道程序系统中,操作系统同时将多个作业保存在内存中。操作系统选择内存中的一个作业开始执行,该作业运行期间,可能必须等待另一个任务(如I/O操作)的完成。对于非多道程序系统,CPU就会空闲等待;对于多道程序系统,操作系统会切换到另一个作业执行。当该作业需要等待时,CPU会再次切换;当第一个作业等待的任务完成后会重新获得CPU。

图5-3(a)给出了单道程序的运行情况,可以看出在 $t_1 \sim t_2$ 、 $t_3 \sim t_4$ 时间段内CPU空闲。在引入多道程序技术后,CPU便可以通过任务切换一直处于忙碌状态,图5-3(b)所示为4道程序的运行情况。

多道程序的特征有如下3点。

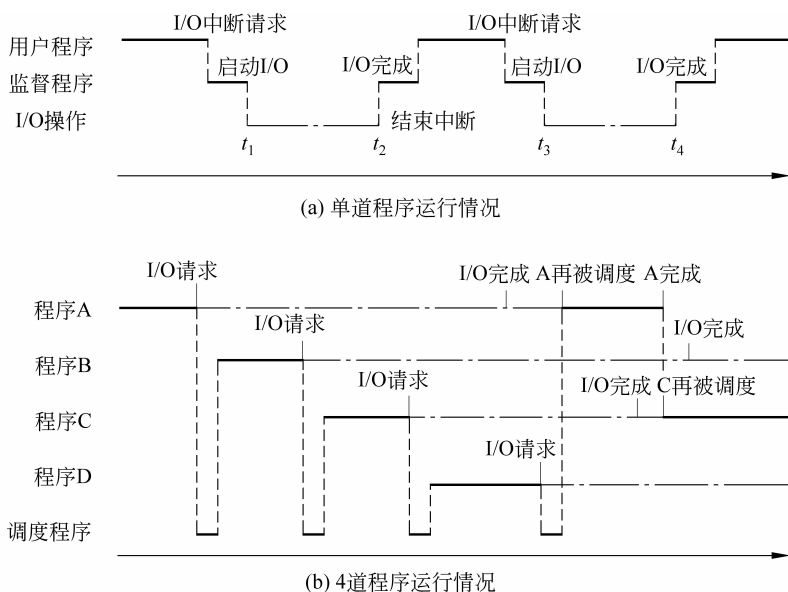


图 5-3 单道和多道程序运行情况

——表示程序占有 CPU - · - 表示程序处于等待中

- (1) 多道。计算机主存中同时存放几道相互独立的程序。
- (2) 宏观上并行。同时进入系统的几道程序都处于运行过程中,即它们都开始运行但都未运行完毕。
- (3) 微观上串行。从微观上看主存中的多道程序轮流或分时占有处理器交替执行。

5.2.4 分时操作系统

分时操作系统(time-sharing system),把处理器的运行时间分成很短的时间片,按时间片轮流把处理器分配给各联机作业使用。若某个作业在分配给它的时间片内不能完成其计算,则该作业暂时中断,把处理器让给另一个作业使用,等待下一轮时再继续其运行。由于计算机速度很快,作业运行轮转得很快,给每个用户的印象是好像它被独占,但事实上它为许多用户所共享。每个用户都可以通过自己的终端向系统发出各种操作控制命令,完成作业的运行。

第一个分时操作系统是在美国麻省理工学院计算机科学系教授费尔南多·考巴托(F. J. Corbató, 1926—)领导下研制完成的兼容分时操作系统(compatible time-sharing system, CTSS)。该系统最初是在 1961 年为 IBM 709 开发的,后来又转移用于 IBM 7094 计算机,可以同时支持 32 个交互式用户。系统向用户提供一组交互式的命令,这样他们就能通过终端来处理文件、编译和运行程序。

与后来的系统相比,CTSS 是相当原始的。该系统运行在一台存储器为 32 000 个 36 位字的机器上,常驻监控程序占用了 5000 个。当控制权被分配给一个交互用户时,该用户的程序和数据被载入到主存储器剩余的 27 000 个字的空间中。程序通常在第 5000 个字单元处开始被载入,这简化了监控程序和内存管理。系统时钟以大约每 0.2s

一个的速度产生中断,在每个时钟中断处,操作系统恢复控制权,并将处理器分配给另一位用户。因此在固定的时间间隔内,当前用户被抢占,另一个用户被载入。为了以后便于恢复,在新的用户程序和数据被读入之前,老的用户程序和数据被写出到磁盘。随后,当获得下一次机会时,老的用户程序代码和数据从磁盘恢复到主存储器中。

为减少磁盘开销,只有当新来的程序需要重写用户存储空间时,用户存储空间才被写出。CTSS 操作原理如图 5-4 所示。假设有 4 个交互用户,其存储器需求如下。

JOB1: 15 000

JOB2: 20 000

JOB3: 5000

JOB4: 10 000

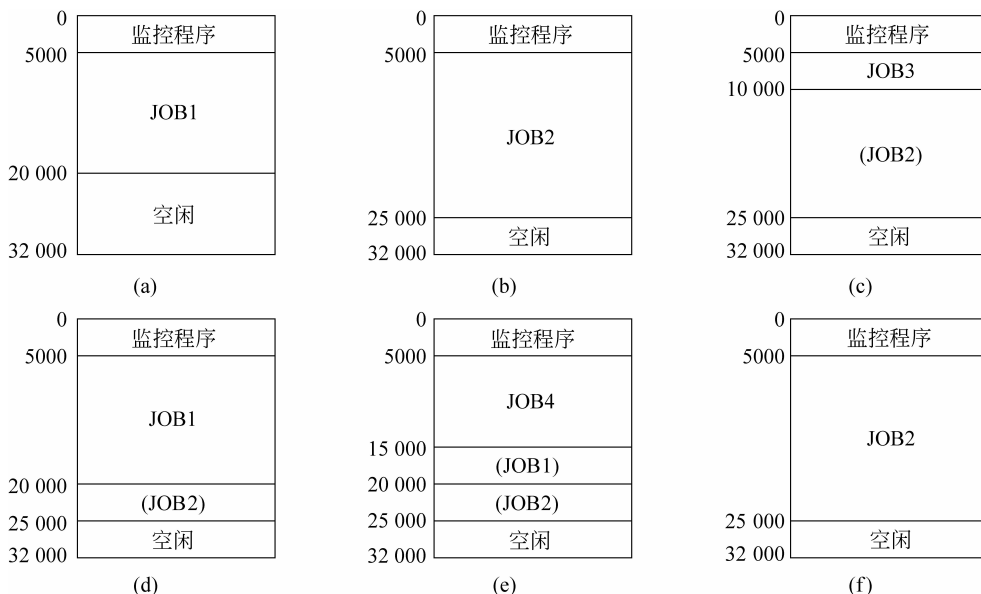


图 5-4 CTSS 操作原理

(a) 最初,监控程序载入 JOB1 并把控制权转交给它。

(b) 稍后,监控程序把控制权转交给 JOB2。由于 JOB2 比 JOB1 需要更多的存储空间,因此 JOB1 必须先被写出,然后载入 JOB2。

(c) 接下来,JOB3 被载入并运行,但由于 JOB3 比 JOB2 小,JOB2 的一部分仍然留在主存储器中,以减少写磁盘的时间。

(d) 稍后,监控程序把控制权交回 JOB1,当 JOB1 载入存储器时,JOB2 的另一部分将被写出。

(e) 当载入 JOB4 时,JOB1 的一部分和 JOB2 的一部分仍留在主存储器中。

(f) 此时,如果 JOB1 或 JOB2 被激活,则只需要载入一部分。在该例中是 JOB2 接着运行,这就要求 JOB4 和 JOB1 留在存储器的那一部分被写出,然后读入 JOB2 的其余部分。

CTSS 试验成功以后,美国国防部的 ARPA 出巨资支持研发第二代分时操作系统,建立了 MULTICS(multiplexed information and computing service)项目,即多路信息计

算系统,简称 MAC 项目。该项目从 1963 年 7 月 1 日开始,由 MIT、通用电气公司(GE)的计算机部以及贝尔实验室 3 家承担了研制任务。后来,贝尔实验室中途退出。1969 年 MAC 项目完成,推出了著名的分时操作系统 MULTICS。其主要特点如下所述:

(1) 首次在大软件的开发中成功地采用了结构化的程序设计方法,使开发周期大大缩短,软件可靠性大大提高。

(2) 成功地采用已有的成熟软件作为工具。MULTICS 中的很大一部分程序是用其自身(即 CTSS)来编写的,这在软件的继承性上是一次成功的尝试。

(3) 全部系统程序是用高级语言 PL/1 编写的,这使系统程序在功能上独立于机器,极大地提高了系统的可移植性。

由于种种原因,MIT 和 GE 公司都没有把 MULTICS 商品化,只有 Honeywell 公司和法国的 Bull 公司在 20 世纪 70 年代初曾推出 MULTICS 的商业版本。但 MULTICS 作为现代操作系统的雏形,它所开创的一系列概念和技术,如内核、进程、层次式目录和面向流的 I/O、把设备当作文件以简化设备管理等,都对后来的操作系统产生了很大的影响,甚至被作为基本技术、核心技术而承袭下来,因而在计算机系统的发展史上占有重要的地位。

5.2.5 实时操作系统

实时(real time)表示及时、即时,而实时操作系统是指系统能及时响应外部事件的请求,在规定的时间内完成对该事件的处理,并控制所有实时任务协调一致地运行。按照任务对截止时间的要求来分,实时操作系统分为硬实时操作系统(hard real time system)和软实时操作系统(soft real time system)两类。

硬实时操作系统必须满足任务对截止时间的要求,保证关键任务按时完成,否则可能出现难以预测的结果。这一目标要求对系统内所有延迟都有限制,从获取存储数据到请求操作系统完成任何操作。这一时间约束要求决定了硬实时操作系统没有绝大多数高级操作系统的功能,这是因为这些功能常常将用户与硬件分开,导致难以估计操作所需时间。因此,硬实时操作系统与分时操作系统的操作相矛盾,两者不能混合使用。

在软实时操作系统中,关键实时任务的优先级要高于其他任务的优先级,且在完成之前能保持其高优先级。与硬实时操作系统一样,需要限制操作系统内核的延迟,实时任务不能无休止地处于等待状态。软实时操作系统可以与其他类型的系统相混合。由于没有安全时间界限的支持,它不能应用在工业控制和机器人等领域。但是,软实时操作系统比硬实时操作系统提供更多的功能,可应用于一些其他领域,如多媒体、虚拟现实等。

实时操作系统的早期代表是 IBM 公司在 1979 年推出的 TPF(transaction processing facility)系统,这是一种在实时环境中运行交易处理应用程序的操作系统,主要面向交易量较大的业务,譬如信用卡公司和航空预定系统。现在,随着嵌入式系统的流行,VxWorks、QNX、RTLinux 等实时操作系统得到了广泛应用。

5.2.6 微机操作系统

操作系统到 20 世纪 80 年代已趋于成熟,随着 VLSI 和计算机体系结构的发展,先后形成了微机操作系统、多处理器操作系统、网络操作系统和分布式操作系统。

配置在微机上的操作系统称为微机操作系统。最早出现的微机操作系统,是在 8 位微机上的 CP/M(control program for microcomputers)。后来随着微机的发展,又相应地出现了 16 位、32 位微机操作系统。可见,微机操作系统可按微机的字长而分成 8 位、16 位和 32 位的微机操作系统,也可把微机操作系统分为单用户单任务操作系统、单用户多任务操作系统和多用户多任务操作系统。

单用户单任务操作系统的含义:只允许一个用户上机,且只允许用户程序作为一个任务运行。这是一种最简单的微机操作系统,主要配置在 8 位微机和 16 位微机上。最有代表性的单用户单任务操作系统是由美国 DR(Digital Research)公司在 1975 年推出的、带有软盘系统的 8 位微机操作系统 CP/M 和微软(Microsoft)公司在 1981 年为 IBM-PC 开发的 MS-DOS 操作系统。

单用户多任务操作系统的含义:只允许一个用户上机,但允许将一个用户程序分为若干个任务,使它们并发执行,从而有效地改善系统的性能。目前,在 32 位微机上所配置的 32 位微机操作系统,大多数是单用户多任务操作系统,其中最具有代表性的是 IBM 公司于 1987 年开发的 OS/2 和微软公司研发的 Windows 操作系统。

多用户多任务的含义:允许多个用户通过各自的终端,同时使用同一台主机,共享主机系统中的各类资源,而每个用户程序又可进一步分为几个任务,使它们并发执行,从而可进一步提高资源利用率,增加系统吞吐量。在大、中、小型机中所配置的都是多用户、多任务操作系统;而在 32 位微机上,也有不少是配置的多用户、多任务操作系统。其中,最有代表性的是贝尔实验室于 1969 年开发的 UNIX 操作系统。UNIX 最初是配置在 DEC 公司的小型机 PDP 上,后来被移植到微型机上。

现在最有影响的两个能运行在微机上的 UNIX 操作系统的变形是 Solaris OS 和 Linux OS。Linux 是 UNIX 的一个变形。最初是由芬兰学生 Linus Torvalds 针对英特尔 80386 开发的。由于源代码公开,因此有很多人通过 Internet 与之合作,使 Linux 的性能迅速提高,其应用范围也日益扩大,相应的源代码也急剧膨胀,此时它已是具有全面功能的 UNIX 操作系统。大量在 UNIX 上运行的软件被移植到 Linux 上,而且可以在主要的微机上运行,如 Intel 80x86 Pentium 等。

5.2.7 多处理器系统

早期的计算机系统基本上都是单处理器系统,20 世纪 70 年代逐步出现了多处理器系统(multi-processor system,MPS),计算机的体系结构发生了改变,相应地也出现了多处理器操作系统。进入 20 世纪 90 年代中后期。功能较强的主机系统和服务器几乎都毫无例外地采用了多处理器系统,处理器的数目可从两个至数千个甚至更多。可以从不同

角度对多处理器系统的结构进行如下分类。从多处理器之间耦合的紧密程度上可以把MPS分为两类。

1) 紧密耦合(tightly coupled)MPS

紧密耦合通常是通过高速总线或高速交叉开关来实现多个处理器之间的互连的。系统中的所有资源和进程都由操作系统实施统一的控制和管理。

2) 松散耦合(loosely coupled)MPS

在松散耦合中,通常是通过通道或通信线路来实现多台计算机之间的互连。每台计算机都有自己的存储器和 I/O 设备。并配置了 OS 来管理本地资源和在本地运行的进程。

根据系统中所用处理器相同与否,可以将多处理器系统分为如下两类。

1) 非对称多处理

在非对称多处理(asymmetric multiprocessing)系统中,把处理器分为主处理器和从处理器两类,每个处理器都有各自特定的任务。主处理器只有一个,其上配置了操作系统,用于管理整个系统的资源,并负责为各从处理器分配任务。从处理器可有多个,它们执行预先规定的任务及由主处理器所分配的任务。在早期的特大型系统中,较多地采用主—从式操作系统。一般来说,主—从式操作系统易于实现,但资源利用率低。

2) 对称多处理

通常在对称多处理(symmetric multiprocessing)系统中,所有的处理器都是相同的。在每个处理器上运行一个相同的操作系统备份,用它来管理本地资源和控制进程的运行,以及各计算机之间的通信。对称多处理意味着所有处理器对等,处理器之间没有主从关系。目前几乎所有现代操作系统,包括 Windows(从 Windows NT 开始)、Solaris、UNIX、OS/2、Linux 等,都支持对称多处理。这种模式的优点是允许多个进程同时运行。例如,当有 n 个 CPU 时,可同时运行 n 个进程而不会引起系统性能的恶化,然而必须小心地控制 I/O,以保证能将数据送至适当的处理器。同时,还必须注意使各 CPU 的负载平衡,以免发生有的 CPU 超载运行,而有的 CPU 则空闲的情况。

5.2.8 网络操作系统

计算机网络可以定义为一些互连的自主计算机系统的集合。所谓自主计算机是指计算机具有独立处理的能力;而互连则是表示计算机之间能够实现通信和相互合作。可见,计算机网络是在计算机技术和通信技术高度发展的基础上相互结合的产物。

1. 网络操作系统的模式

网络操作系统具有以下两种工作模式。

1) 客户/服务器(client/server,C/S)模式

该模式是在 20 世纪 80 年代发展起来的,目前仍广为流行的网络工作模式。网络中的站点可分为以下两大类。

(1) 服务器。它是网络的控制中心,其任务是向客户提供一种或多种服务。服务器