

第 2 章

操作系统基础知识

2.1 备考指南

2.1.1 考纲要求

根据考试大纲中相应的考核要求，在“操作系统基础知识”模块中，要求考生掌握以下几方面的内容。

- (1) 操作系统基础知识，包括操作系统的类型、功能。
- (2) 处理机管理，包括进程的基本概念，进程的控制，进程间的通信，进程调度，信号量与 P、V 操作，高级通信原语，死锁和线程的基本概念等。
- (3) 存储管理，包括主存保护、分区存储管理、分页存储管理、分段存储管理和虚存管理等。
- (4) 设备管理，包括设备的类型、与设备分配有关的调度算法、通道、DMA 与缓冲技术、假脱机和磁盘调度等。
- (5) 文件管理，包括文件与文件系统的概念、文件的结构和组织等。
- (6) 作业管理，包括作业管理的基本概念、作业调度及调度算法、评价作业调度算法应用的目的及对系统性能的影响。
- (7) 图形用户界面和操作方法。

2.1.2 考点统计

“操作系统基础知识”模块在历次程序员考试试卷中出现的考核知识点及分值分布情况如表 2-1 所示。



表 2-1 历年考点统计表

年 份	题 号	知 识 点	分 值
2017 年 下半年	上午题: 25~27	信号量、存储管理、PV 操作等	3 分
	下午题: 无		0 分
2017 年 上半年	上午题: 25~27	信号量、进程的三态模型、存储管理等	3 分
	下午题: 无		0 分
2016 年 下半年	上午题: 24~27	操作系统基础知识、状态转换、信号量、存储管理等	4 分
	下午题: 无		0 分
2016 年 上半年	上午题: 23~27	操作系统基础知识、信号量等	5 分
	下午题: 无		0 分
2015 年 下半年	上午题: 26~27	进程的三态模型、信号量等	2 分
	下午题: 无		0 分

2.1.3 命题特点

纵观历年试卷,本章知识点是以选择题的形式出现在试卷中的。本章知识点在历次考试的“综合知识”试卷中,所考查的题量为 2~3 道选择题,所占分值为 2~3 分(约占试卷总分值 75 分中的 2.67%~4.00%)。

2.2 考点串讲

2.2.1 操作系统概述

一、操作系统的定义

操作系统是计算机系统最重要的系统软件,其他所有的软件都是建立在操作系统之上的,并在操作系统的统一管理和支持下运行。任何用户都是通过操作系统使用计算机的。

操作系统的定义为:操作系统(Operating System, OS)是计算机系统中的一个系统软件,它管理和控制计算机系统的硬件和软件资源,合理地组织计算机工作流程,以便有效地利用这些资源为用户提供一个功能强大、使用方便的工作环境,从而在计算机与用户之间起到接口的作用。

操作系统的主要任务是使硬件所提供的能力得到充分的利用,支持应用程序的运行并提供相应的服务。由于操作系统在计算机系统中占据着重要地位,所以它已经成为现代计



算机系统中一个必不可少的关键组成部分。

二、操作系统的作用

(1) 通过资源管理,提高工作效率。

操作系统的主要作用就是通过 CPU 管理、存储管理、设备管理和文件管理,对各种资源进行合理的分配,改善资源的共享和利用程度,最大限度地发挥计算机系统的工作效率,提高计算机系统的“吞吐量”(即系统在单位时间内处理工作的能力)。

(2) 改善人机界面,提供友好的工作环境。

操作系统既是计算机硬件和各种软件之间的接口,又是用户与计算机之间的接口。安装操作系统后,用户面对的不再是笨拙的裸机、由 0 和 1 组成的代码及一些难懂的机器指令,而是操作便利、服务周到的操作系统,操作系统明显地改善了用户界面,提高了用户的工作效率。

三、操作系统的特征

操作系统主要有并发性、共享性、虚拟性和不确定性四个基本特征。

1) 并发性(Concurrency)

并发性是指在计算机系统中存在着许多同时进行的活动。对计算机系统而言,并发是指宏观上看系统内有多道程序同时运行,微观上看实际上是串行运行。

2) 共享性(Sharing)

共享性是指系统中各个并发活动要共享计算机系统中的各种软、硬件资源,因此操作系统必须解决在多道程序间合理地分配和使用资源。

3) 虚拟性(Virtual)

虚拟性是操作系统中的重要特征,所谓虚拟是指把物理上的一台设备变成逻辑上的多台设备。例如,我们将在本章后面介绍的假脱机(Spooling)技术,就是利用快速、大容量、可共享的磁盘作为中介,模拟多个非共享的低速的输入/输出设备,这样的设备称为虚拟设备。

4) 不确定性(Non-Determinacy)

通常一个程序的初始条件相同时,无论何时运行,结果都应该相同。但由于操作系统并发执行系统内的各种进程,与这些进程有关的事件如从外部设备来的中断、输入输出请求、各种运行故障、发生的时间等都不可预测,如果处理不当,将导致系统出错,这种不确定性所带来的错误是很难查找的。

四、操作系统的功能

1) 处理机管理

处理机是计算机系统的核心,在单用户系统或单道系统中,处理机为一个用户或一个作业服务,其管理简单,但资源利用率低。为提高系统资源的利用率,引入了多道程序技术,即多个程序(作业)同时运行。在多道程序或多用户的情况下,要组织多个作业同时运行,对多个用户进行响应,就需要解决对处理机的分配、调度和资源回收等问题。处理机管理负责解决如何把 CPU 时间合理地、动态地分配给程序运行的基本单位——进程,使处理机





得到充分的利用。许多操作系统是以作业和进程的方式进行管理的,实现作业和进程的调度,分配处理机,控制作业和进程的执行。现代的操作系统还引入了线程(Thread)作为分配处理机的基本单位。

由于操作系统对处理机的管理策略不同,其提供的作业处理方式也就不同,如批处理方式、分时处理方式和实时处理方式,从而呈现在用户面前的就有不同的操作系统。在操作系统中,最重要的资源是处理机,最重要的管理是处理机管理。

2) 存储管理

计算机系统中,存储器(一般称为主存或内存)是运行程序和存放工作数据的部件,存储管理的工作主要是对内存存储器进行分配、扩充和保护。

- 内存分配:在内存中除了操作系统和其他系统软件外,还要有一个或多个用户程序。如何分配内存,以保证系统及各用户程序的存储区互相不冲突,是内存分配所要解决的问题。
- 存储保护:系统中有多个程序在运行,如何保证一道程序在执行过程中不会有意或无意地破坏另一道程序?如何保证用户程序不会破坏系统程序?这些就是存储保护问题。
- 内存扩充:当用户作业所需要的内存量超过计算机系统所提供的内存容量时,如何把内部存储器和外部存储器结合起来管理,为用户提供一个容量比实际内存大得多的虚拟存储器,使这个虚拟存储器和内存一样方便使用,这就需要使用内存扩充。

存储器是计算机系统最重要的资源之一,因为任何程序和数据,以及各种控制用的数据结构,都必须占有一定的存储空间,因此,存储管理的目的就是尽量提高内存的使用效率。存储管理的好坏直接影响着系统性能。

3) 设备管理

现代计算机系统常常配置很多种类的输入/输出设备,它们的输入/输出速度差别很大。计算机系统常常采用通道、控制器和设备三级控制方法管理这些设备。设备管理的任务就是监视这些资源的使用情况,根据一定的分配策略,把通道、控制器和设备分配给请求输入/输出操作的程序,并启动设备完成所需的操作。为了发挥设备和处理机的并行工作能力,常常采用缓冲技术和虚拟技术。

由于输入/输出设备种类很多,使用方法各不相同,因此,设备管理应为用户提供一个良好的界面,使具体的设备特性透明化,以使用户能方便、灵活地使用这些设备。

4) 文件管理(信息管理)

文件管理是对系统软件资源的管理。对用户来说,文件系统是操作系统中最直观的部分。我们把程序和数据统称为信息或文件,当一个文件暂时不用时,就把它放到外部存储器(如磁盘、磁带和光盘等)上保存起来。对这些文件如果不能很好地进行管理,就会引起混乱,甚至使其遭受破坏。这就是文件管理需要解决的问题。

文件管理的功能包括:建立、修改和删除文件;按文件名进行访问;决定文件信息的存放位置、存放形式及存取权限;管理文件间的联系及提供对文件的共享、保护和保密等,允许多个用户协同工作又不引起混乱。



5) 用户接口(作业管理)

上述四项功能是操作系统对软、硬件资源的管理。除此以外,操作系统也必须为用户提供一个友好的用户接口——命令接口和图形接口。一般来说,用户通过两种命令接口请求操作系统的服务。一种用户接口是作业一级的接口,即提供一组控制操作命令,如 UNIX 的 Shell 命令语言或作业控制语言(JCL)让用户组织和控制自己作业的运行。作业控制又分成两类:联机控制和脱机控制。另一种用户接口是程序一级的接口(编程接口),即提供一组广义指令(或称系统调用、程序请求)供用户程序和其他系统程序调用。当这些程序要求进行数据传输、文件操作或有其他资源要求时,通过这些广义指令向操作系统提出申请,并由操作系统代为完成。

操作系统对计算机的资源进行全面管理,它的基本特征是多任务并行和多用户资源共享。多任务并行是指操作系统可以支持用户同时提交多项任务,同时工作;资源共享是指系统中的资源为多个用户共同使用。

五、操作系统的类型

根据操作系统的使用环境和对作业的处理方式来划分,操作系统主要有以下几种基本类型。

1) 批处理操作系统

在批处理操作系统(Batch Processing Operating System)中,系统操作员将作业成批提交,由操作系统选择作业调入内存加以处理,最后由操作人员将运行结果交给用户。

批处理系统的特点:一是“多道”,是指系统内可同时容纳多个作业;二是“成批”,是指系统成批自动运行多个作业。批处理系统的目标是提高资源利用率和实现作业执行的自动化。

批处理操作系统分为单道批处理和多道批处理两种。

- 单道批处理操作系统:一次可提交多个作业,而不是单个作业。当一个作业运行结束后,随即自动调入同批的下一个作业运行,从而节省了作业之间的人工操作时间,提高了资源的利用率。早期单道批处理系统解决了作业自动转换的问题,从而减少了作业建立和人工操作的时间。单道批处理存在的主要问题是:CPU 和 I/O 设备使用忙闲不均(取决于当前作业的特性),对以计算为主的作业,外设空闲;对以 I/O 为主的作业,CPU 空闲。
- 多道批处理操作系统:正是为了解决单道批处理操作系统存在的问题而产生了多道批处理操作系统。它除了保持作业自动转换的功能外,还能支持同一批中的多道用户程序在一个 CPU 上同时运行。作业调度程序从后备作业中选取多个作业进入主存,在任意一个时刻,每当运行中的一个作业因输入/输出操作而需要调用外部设备时,就把 CPU 及时交给另一道等待运行的作业,从而将主机与外部设备的工作方式由串行改为并行,进一步避免了因主机等待外设完成任务而白白浪费宝贵的 CPU 时间的情况。

2) 分时操作系统

分时操作系统(Time Share Operating System)是指一台计算机连接多个终端,系统把 CPU 时间分为若干个时间片,采用时间片轮转的方式处理用户的服务请求,对每个用户能保证





及时响应,并提供交互会话能力。

分时操作系统具有下述特点。

- 多用户同时性:允许多个用户同时联机使用计算机。
- 交互性:每个用户可随时通过终端向系统提出服务请求,系统也可随时通过终端响应用户,从而加快了调试过程。
- 独立性:由于采用时间片轮转方式使一台计算机同时为多个用户服务,对于每个用户的操作命令又能快速响应,因此,用户彼此之间都感觉不到别人也在使用同一台计算机,如同自己独占计算机一样。
- 及时性:系统对用户的响应非常及时,不会让用户等待执行命令的处理时间过长。

分时操作系统的主要目标是保证用户响应的及时性。通常,计算机系统中往往同时采用批处理和分时处理方式来为用户服务,即时间要求不强的作业放入“后台”(批处理)处理,需频繁交互的作业放在“前台”(分时)处理。

3) 实时操作系统

实时操作系统(Real Time Operating System)是随着计算机应用于实时控制和实时信息处理而发展起来的。实时操作系统是指系统能够及时响应事件,并以足够快的速度完成对该事件的处理。实时操作系统包括实时控制系统和实时处理系统。实时控制系统是指生产过程控制(如炼钢、电力生产和数控机床)及武器控制等;实时处理系统是指实验数据采集和订票系统等。

实时操作系统的主要特点是及时性和高可靠性。

4) 网络操作系统

网络操作系统(Network Operating System)开发是在原来各自计算机操作系统的基础上,按照网络体系结构的协议、标准进行开发的,包括计算机网络管理、通信、资源共享、系统安全和多种网络应用服务等。其功能主要包括高效、可靠的网络通信;对网络中共享资源的有效管理;电子邮件、文件传输、共享硬盘、打印机等服务;网络安全管理;互操作能力。

5) 分布式操作系统

分布式操作系统(Distributed Operating System)与网络操作系统都是工作在一个由多台计算机组成的系统中,这些计算机之间可以通过一些传输设备进行通信和系统资源共享。分布式操作系统更倾向于任务的协同执行,并且各系统之间无主次之分,系统之间也无须采用标准的通信协议进行通信。分布式操作系统基本上废弃(或改造)了各单机的操作系统,整个网络设有单一的操作系统,由这个操作系统负责整个系统的资源分配和调度,为用户提供统一的界面。用户在使用分布式操作系统时不需要像使用网络操作系统那样,指明资源在哪台计算机上,因此分布式操作系统的透明性、稳固性、统一性及系统效率都比网络操作系统要强,但实现起来难度也大。分布式操作系统对于多机合作和系统重构、稳固性和容错能力有更高的要求,希望分布式操作系统有更短的响应时间、更大的吞吐量和更高的可靠性。

分布式操作系统与网络操作系统最大的差别是:网络操作系统的用户必须知道网址,而分布式操作系统的用户则不必知道计算机的确切地址;分布式操作系统负责全系统的资源分配,通常能很好地隐藏系统内部的实现细节,如对象的物理位置、并发控制、系统故



障处理等对用户都是透明的。

6) 微机操作系统

微机操作系统(Microcomputer Operating System)是指配置在微型计算机上的操作系统。常用的微机操作系统有 DOS、Windows、OS/2、UNIX 和 Linux 等。其中, Microsoft 公司开发的单用户单任务操作系统 DOS 是首先在 IBM-PC 机上使用的微机操作系统。MS-DOS 操作系统是 16 位微机单用户单任务操作系统的标准。多任务操作系统 Windows 98/NT/2000/XP 是 Microsoft 公司开发的一系列图形用户界面的多任务、多线程的操作系统。

7) 嵌入式操作系统

嵌入式操作系统(Embedded Operating System)运行在嵌入式智能芯片环境中, 对整个智能芯片及其控制的各种部件和装置等资源进行统一协调、处理、指挥和控制。

六、研究操作系统的观点

研究和分析操作系统, 可以从资源管理观点和虚拟机观点出发。

1) 资源管理观点

引入操作系统是为了合理地组织计算机的工作流程, 管理和分配计算机系统硬件和软件资源, 使资源能为多个用户共享。因此, 操作系统是计算机资源的管理者。

这里的资源是指计算机系统数值计算和数据处理所需的物质基础, 通常分为系统硬件资源和软件资源。硬件资源是组成计算机和计算机操作所需的物理实体, 它们是看得见、摸得着的设备, 如处理机、存储器及输入/输出设备(键盘、显示器、打印机和磁盘等)。软件资源是依赖于一定的物理实体才能被人们所感知的一类资源, 如程序和数据等, 它们可经显示器或打印机等设备展现给用户。操作系统是控制和管理计算机系统资源的一组程序, 其工作是当用户程序和其他程序争用这些资源时提供有序的和可控的分配。

我们通常将操作系统分为 CPU 管理、存储管理、设备管理、文件管理、用户与操作系统接口五个主要部分。它主要研究资源的使用情况、资源的分配策略及分配和回收资源。

2) 虚拟机观点

从服务用户的机器扩充的观点来看, 操作系统为用户使用计算机提供了许多服务功能和良好的工作环境。用户不再直接使用硬件机器(称为裸机), 而是通过操作系统来控制和使用计算机, 从而把计算机扩充为功能更强、使用更方便的计算机系统(称为虚拟计算机)。操作系统的全部功能, 如系统调用、命令、作业控制语言等, 称为操作系统虚拟机。

虚拟机观点从功能分解的角度出发, 考虑操作系统的结构, 将操作系统分成若干个层次, 每一层次完成特定的功能, 从而构成一个虚拟机, 并为上一层次提供支持, 构成它的运行环境。这样, 通过逐个层次的功能扩充最终完成操作系统虚拟机, 从而向用户提供各种服务, 完成用户的各项任务。

2.2.2 进程管理

一、基本概念

在计算机系统上运行的程序是指令的集合, 每一个程序完成特定的任务。在只允许一个程序运行的系统(称为单道系统)中, 这个程序独占系统资源, 而系统按程序的指令顺序运





行,程序的顺序执行有两个基本特征:封闭性和可再现性。

- 封闭性:指程序运行时独占系统资源,只有程序本身能改变系统的状态。
- 可再现性:指程序运行不受外部因素的影响,只要初始条件相同,运行结果就相同。

多道程序系统让多个程序在系统中轮流运行,当一个程序不用处理机时,另一个程序就使用。也就是说,处理机在程序间来回切换,从而获得宏观上的并行(微观上的串行),以提高处理机的利用率。这种切换,通常是由中断引起的。由于中断以不可预测的次序发生,即程序的指令执行序列也以不可预测的次序前进,这样就会产生操作系统的另一个特性——不确定性。即在多道程序系统中,顺序程序的封闭性和可再现性消失了,需要采用一个新的概念——进程来描述程序的执行。进程是运行中的程序,是系统进行资源分配和调度的独立单位。

1. 进程及其组成

进程是一个程序关于某个数据集的一次运行。进程是一个动态的概念,而程序是静态的概念,是指令的集合。因此,进程具有动态性和并发性。

进程通常由程序、数据和进程控制块(PCB)组成。程序是进程运行所对应的运行代码,一个进程对应于一个程序,一个程序可以同时对应于多个进程,代码在运行过程中不会被改变的程序,常称为纯码程序或可重入程序,这类程序是可共享的程序。

进程控制块是进程动态特性的集中反映,也是进程存在的唯一标志。在操作系统中,进程是进行系统资源分配、调度和管理的最小单位。现代操作系统中还引入了线程,线程是比进程更小的、能独立运行的基本单位,在引入线程的操作系统中,线程是进程中的一个实体,是CPU调度和分派的基本单位,是处理机分配的最小单位。

2. 进程的状态及其转换

在多道系统中,进程的运行是走走停停的,在处理机上的交替运行,使它的运行状态不断变化。进程的状态主要有三态模型和五态模型。三态模型中最基本的状态有三种:运行、就绪和阻塞。

- 运行(running):正占用处理机。
- 就绪(ready):只要获得处理机即可运行。
- 阻塞(blocked):也称等待或挂起状态,正等待某个事件(如I/O完成)的发生。

在进程运行的过程中,由于自身进展情况及外界环境的变化,这三种基本状态可以在一定的条件下相互转换,进程的状态及转换如图2-1所示。

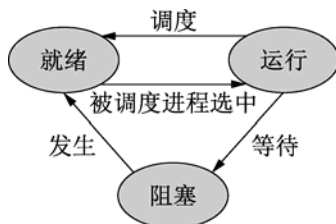


图 2-1 进程状态及其转换

五态模型比三态模型更加复杂,在三态模型的基础上增加了新建态和终止态。新建态对应于进程刚刚被创建还没有被提交时的状态,此时应在等待系统完成创建进程的所有必



要信息。创建进程时分两个阶段：第一个阶段为一个新进程创建必要的管理信息；第二个阶段让该进程进入就绪状态。有了新建态，操作系统往往因系统的性能和内存容量的限制推迟新建态进程的提交。进程的终止态也可分为两个阶段：第一个阶段等待操作系统进行善后处理；第二个阶段释放内存。

二、进程的控制

进程的控制就是对系统中所有进程从创建到消亡的全过程实施有效的控制。不仅要控制正在运行的进程，而且还要能创建新的进程，撤销已完成的进程。进程的控制机构是由操作系统内核实现的。通常将与硬件密切相关的模块放在紧挨硬件的软件层中，并使它们常驻内存，以便提高操作系统的运行效率，通常将这部分称为操作系统的内核，它为系统对进程进行控制和对存储器进行管理提供了有效的控制机制。

1. 支撑功能

1) 中断处理

操作系统的各种重要活动最终都依赖于中断。例如，各种类型的系统调用、键盘命令的输入、设备驱动及文件系统等都依赖于中断。通常内核只对中断进行“有限次处理”，然后转入有关进程继续处理。这不仅可以减少中断处理的时间，还可以提高程序的并发性。

2) 时钟管理

操作系统的许多活动要用到时钟管理。如在分时系统时间片调度算法中，当时间片用完时，由时钟管理产生一个中断信号，通知调度程序重新调度。在实时系统中的截止时间控制、批处理系统中的最长运行时间的控制等都要用到时钟管理。

3) 原语操作

内核在执行某些基本操作时，往往是通过原语操作来实现的。原语是由若干条机器指令构成的，用于完成特定功能的一段程序。原语在执行的过程中是不可分割的。进程控制原语主要有：创建原语、撤销原语、挂起原语、激活原语、阻塞原语以及唤醒原语。

2. 资源管理功能

资源管理功能包括：进程管理、存储器管理和设备管理。

三、进程间的通信

1. 同步与互斥

在操作系统中，多个进程并发执行，因此进程间必然存在资源共享和相互合作的问题。

1) 进程间的同步

一般情况下，一个进程相对于另一个进程的速度是不可预测的，也就是说，进程之间是异步运行的。为了成功地协同工作，有关进程在某些确定的点上应当保持同步：一个进程到达了这些点后，除非另一进程已经完成了某个活动，否则就停下来，等待该活动结束。

同步是指进程之间的一种协同工作关系，使这些进程相互合作，共同完成一项任务。进程间的直接相互作用构成进程的同步。同步机制应满足的基本要求是：有描述能力，可以实现，效率高，使用方便。

2) 进程间的互斥

在多道系统中，各进程可以共享各类资源，但有些资源却一次只能供一个进程使用。



这种资源称为临界资源,如打印机、公共变量、表格等。互斥是要保证临界资源在某一时刻只被一个进程访问。

3) 临界区管理的原则

临界区是进程中对临界资源实施操作的那段程序。对互斥临界区管理的原则是:有空即进,无空则等,有限等待,让权等待。

2. 信号量机制

信号量机制是一种有效的进程同步与互斥工具,目前主要有:整型信号量、记录型信号量、信号量集机制。

1) 整型信号量与 P 操作和 V 操作

信号量是一个整型变量,根据控制对象的不同被赋予不同的值。信号量分为两类:公用信号量,实现进程间的互斥,初值为 1 或资源的数目;私用信号量,实现进程间的同步,初值为 0 或某个正整数。

信号量 S 的物理意义: $S \geq 0$ 表示某资源的可用数,若 $S < 0$,则其绝对值表示阻塞队列中等待该资源的进程数。

除了设置初值外,对信号量只能进行特殊的操作: P 操作和 V 操作。P 操作和 V 操作都是不可分割的原子动作,也称为原语,其中 P 操作表示申请一个资源, V 操作表示释放一个资源。

P 操作和 V 操作都是原语。利用信号量 S 的取值表示共享资源的使用情况。在使用时,把信号量 S 放在进程运行的环境中,赋予其不同的初值,并在其上实施 P 操作和 V 操作,以实现进程间的同步与互斥。

P 操作和 V 操作的定义如下。

P(S): ① $S = S - 1$; ② 若 $S < 0$, 则该进程进入 S 信号量的队列中等待。

V(S): ① $S = S + 1$; ② 若 $S \leq 0$, 则释放 S 信号量队列上的一个等待进程,使之进入就绪队列。

当 $S > 0$ 时,表示还有资源可以分配;当 $S < 0$ 时,其绝对值表示信号量等待队列中进程的数目。每执行一次 P 操作,意味着要求分配一个资源;每执行一次 V 操作,就意味着释放一个资源。

2) 利用 P 操作和 V 操作实现进程的互斥

令信号量 `mutex` 的初值为 1,进入临界区时执行 P 操作,退出临界区时执行 V 操作,于是临界区就改写成下列形式的代码段:

```
P(mutex);  
临界区  
V(mutex);
```

由于 `mutex` 初值为 1, P、V 是原子操作,可以实现互斥。

3. 高级通信原语

P 操作和 V 操作是用来协调进程间关系的,编程较困难、效率低,而且没有信息交换,故常称为低级通信原语。交换的信息量多时要引入高级通信原语,进程高级通信的类型主要有如下几种。



- 共享存储系统：相互通信的进程共享某些数据结构或存储区，以实现进程之间的通信。
- 消息传递系统：进程间的数据交换以消息为单位，程序员直接利用系统提供的一组通信命令(原语)来实现通信。如 `Send(A)`、`Receive(A)`。
- 管道通信：所谓管道，是指用于连接一个读进程和一个写进程，以实现它们之间通信的共享文件(pipe 文件)。向管道(共享文件)提供输入的发送进程(即写进程)，以字符流的形式将大量的数据送入管道；而接收进程可从管道接收大量的数据。由于通信是采用管道的方式，所以称为管道通信。

四、进程调度

进程调度即处理机调度，它的主要功能是确定在什么时候分派处理机，并确定分给哪一个进程。在一些操作系统中，一个作业从提交到完成需要经历高、中、低三级调度。

- 高级调度：又称“长调度”“作业调度”或“接纳调度”，它决定处于输入池中的哪个后备作业可以调入主系统，以便做好运行的准备，成为一个或一组就绪进程。系统中一个作业只需经过一次高级调度。
- 中级调度：又称“中程调度”或“对换调度”，它决定处于交换区中的哪个就绪进程可以调入内存，以便直接参与对 CPU 的竞争。在内存资源紧张时，为了将进程调入内存，必须将内存中处于阻塞状态的进程调至交换区，以便为调入进程腾出空间。
- 低级调度：又称“短程调度”或“进程调度”，它决定处于内存中的哪个就绪进程可以占用 CPU，它是操作系统中最活跃、最重要的调度程序，对系统的影响很大。

1. 调度方式

调度方式是指当有更高优先级的进程到来时该如何分配 CPU。调度方式分为可剥夺式和不可剥夺式两种。可剥夺式是指当有更高优先级的进程到来时，强行将正在运行的进程所占用的 CPU 分配给高优先级的进程；不可剥夺式是指当有更高优先级的进程到来时，必须等待正在运行的进程自动释放占用的 CPU，然后将 CPU 分配给高优先级的进程。

2. 进程调度算法

常用的进程调度算法有：先来先服务、时间片轮转、优先级调度和多级反馈调度。

1) 先来先服务

先来先服务(FCFS)是按照作业提交或进程变为就绪状态的先后次序，分配 CPU。即每当进入进程调度时，总是将就绪队列队首的进程投入运行。FCFS 的特点比较有利于长作业，而不利于短作业；有利于 CPU 繁忙的作业，而不利于输入/输出繁忙的作业。

2) 时间片轮转

FCFS 算法主要用于宏观调度，时间片轮转算法主要用于微观调度，通过时间片轮转，提高进程并发性和响应时间，从而提高资源利用率。

时间片轮转的实现过程是将系统中所有的就绪进程按照 FCFS 原则，排成一个队列。每次调度时将 CPU 分派给队首进程，让其执行一个时间片。时间片的长度从几毫秒到几百毫秒。在一个时间片结束时，发生时钟中断，调度程序据此暂停当前运行进程的执行，将其



送到就绪队列的末尾,并通过上下文切换执行当前的队首进程。进程可以未使用完一个时间片,就出让 CPU(如阻塞)。

时间片长度的确定主要考虑以下四个方面。

- 时间片长度变化的影响:时间片过长,退化为 FCFS 算法,进程在一个时间片内都执行完,造成响应时间长;时间片过短,用户的一次请求需要多个时间片才能处理完,上下文切换次数增加,系统效率降低,同样造成响应时间增长。
- 对响应时间的要求: $T(\text{响应时间}) = N(\text{进程数目}) \times q(\text{时间片})$ 。
- 就绪进程的数目:数目越多,时间片越小。
- 系统的处理能力:应当使用户输入在一个时间片内能处理完,否则会使响应时间、平均周转时间和平均带权周转时间延长。

3) 优先级调度

优先级调度分为静态优先级和动态优先级两种。

- 静态优先级:进程的优先级是在创建时就已确定好了,直到进程终止都不会改变。确定优先级的依据主要有进程类型(系统进程优先级较高)、对资源的需求(对 CPU 和内存需求较少的进程优先级较高)、用户要求(紧迫程度和付费多少)。
- 动态优先级:在创建进程时赋予一个优先级,在进程运行过程中还可以改变,以便获得更好的调度性能。进程每执行一个时间片,就降低其优先级,从而一个进程持续执行时,其优先级可能会降低到出让 CPU 为止。

4) 多级反馈调度

多级反馈调度算法是时间片轮转算法和优先级算法的综合与发展。其优点是:照顾了短进程,提高了系统吞吐量,缩短了平均周转时间;照顾输入/输出型进程,获得较好的输入/输出设备利用率和缩短响应时间;不必估计进程的执行时间,动态调节优先级。

五、死锁

1. 死锁的基本概念

当若干个进程竞争使用资源时,可能每个进程要求的资源都被另一进程占用,于是也就没有一个进程能继续运行,这种情况称为死锁。例如, P1 进程占有资源 R1, P2 进程占有资源 R2,这时, P1 又需要资源 R2, P2 也需要资源 R1,它们在等待对方占有的资源时,又不会释放自己占有的资源,因而使双方都进入了无限等待状态。死锁是系统的一种出错状态,不仅浪费大量的系统资源,甚至还会导致整个系统的崩溃,所以死锁是应该尽量预防和避免的。

系统发生死锁时,死锁进程的个数至少为两个,所有死锁进程都有等待资源,其中至少有两个进程已占有资源。产生死锁的情况主要有:进程推进顺序不当;同类资源分配不当;PV 操作使用不当。

2. 产生死锁的四个必要条件

产生死锁的原因:一是系统提供的资源数量有限,不能满足每个进程的使用;二是多道程序运行时,进程推进顺序不合理。产生死锁必须同时具备下述四个条件。

- 互斥:进程互斥使用资源,任意时刻一个资源只为一个进程所独占,其他进程若请求一个已被占用的资源,只能等待占用者释放后才能使用。



- 不可剥夺(不可抢占): 进程所获得的资源在未使用完毕之前, 不能被其他进程强行剥夺, 而只能由获得该资源的进程自己释放。
- 请求保持: 进程每次申请它所需要的一部分资源, 在申请新的资源的同时, 继续占用已分配到的资源。零星地请求资源, 即已获得部分资源后再次请求资源时被阻塞。
- 循环等待: 在进程资源有向图中存在一个进程环路, 环路中每一个进程已获得的资源同时被下一个进程所请求。

进程资源有向图由方框、圆圈和有向边三个部分组成。其中, 方框表示资源, 圆圈表示进程。请求资源: $\bigcirc \rightarrow \square$, 箭头由进程指向资源; 分配资源: $\bigcirc \leftarrow \square$, 箭头由资源指向进程。

3. 解决死锁的方法

解决死锁的方法如下。

- 死锁的预防: 根据产生死锁的四个必要条件, 只要使其中之一不能成立, 死锁就不会出现。
- 死锁的避免: 最著名的死锁避免算法是 Dijkstra 提出的银行家算法。
- 死锁的检测: 采用合理的死锁检测算法确定死锁的存在, 并识别出与死锁有关的进程和资源, 以供系统采用适当的解除死锁的措施。
- 死锁的解除: 检测到死锁发生后, 常采用资源剥夺法和撤销进程法解除死锁。

六、线程

1. 线程的基本概念

线程是比进程更小的能独立运行的基本单位。在引入线程的操作系统中, 线程是进程中的一个实体, 是 CPU 调度和分派的基本单位。线程自己基本上不占用系统资源, 只占用一点儿在运行中必不可少的资源(如程序计数器、一组寄存器和栈), 但它可与同属一个进程的其他线程共享该进程所占用的全部资源。相应的, 线程也同样有就绪、等待和运行三种基本状态。在有的系统中, 线程还有终止状态。

2. 线程的属性

线程的属性如下。

- 每个线程都有一个唯一的标识符和一张线程描述表。
- 不同的线程可以执行相同的程序。
- 同一进程中的各个线程共享该进程的内存地址空间。
- 线程是处理机的独立调度单位, 多个线程是可以并发执行的。
- 线程在生命周期内会经历等待状态、就绪状态和运行状态等各种状态变化。

3. 引入线程的好处

传统的进程有两个基本属性: 可拥有资源的独立单位、可独立调度和分配的基本单位。由于在进程的创建、撤销和切换中, 系统必须为之付出较大的时空开销, 因此在系统中所设置的进程数目不宜过多, 进程切换的频率不宜太高, 这就限制了并发程度的提高。引入线程后, 将传统进程的两个基本属性分开, 将线程作为调度和分配的基本单位, 而将进程



作为独立分配资源的单位。用户可以通过创建线程来完成任务,以减少程序并发执行时付出的时空开销。

引入线程的好处主要有如下几个方面。

- 创建一个新线程花费的时间少。
- 两个线程间切换花费的时间少。
- 由于同一进程内的线程共享内存和文件,线程之间相互通信无须调用内核,故不需要额外的通信机制,使通信更简便,信息传送速度也更快。
- 线程能独立执行,能充分利用和发挥处理机与外围设备并行工作的能力。

2.2.3 存储管理

一、基本概念

现代计算机系统存储系统通常是多级存储体系,至少有主存(内存)和辅存(外存)两级,有的系统有更多级。系统中主存的使用一般分成两部分:一部分为系统空间,存放操作系统本身及相关的系统数据;另一部分为用户空间,存放用户的程序和数据。提高主存的利用率,对主存信息实现有效保护是存储器管理的主要任务。

1. 存储器的结构

存储器的功能是保存数据,存储器的发展方向是高速、大容量和小体积。一般存储器的结构有“寄存器—主存—外存”结构或“寄存器—缓存—主存—外存”结构。下面介绍几个与存储器相关的概念。

(1) 虚拟地址:数据的存放地址是由符号决定的,故又称为符号名地址,或者称为名地址,而把源程序的地址空间称为符号名地址空间或者名空间。它从 0 号单元开始编址,并顺序分配所有的符号名所对应的地址单元,所以它不是主存中的真实地址,故称为相对地址、程序地址、逻辑地址或虚拟地址。

(2) 地址空间:程序中由符号名组成的空间称为地址空间。源程序经过汇编或编译后再经过链接编辑程序加工形成程序的装配模块,即转换为相对地址编址的模块,它是以 0 为基址顺序进行编址的。相对地址也称为逻辑地址或虚拟地址,把程序中由相对地址组成的空间称为逻辑空间。相对地址空间通过地址重定位机构转换到绝对地址空间,绝对地址空间也称为物理地址空间。

(3) 存储空间:简单来说,逻辑地址空间(简称地址空间)是逻辑地址的集合,物理地址空间(简称存储空间)是物理地址的集合。

2. 地址重定位

地址重定位是指程序的逻辑地址被转换成主存的物理地址的过程。在可执行文件装入时需要解决可执行文件中地址(指令和数据)和主存地址的对应关系。由操作系统中的装入程序 Loader 和地址重定位机构来完成。地址重定位分为静态地址重定位和动态地址重定位。

(1) 静态地址重定位,是指在程序装入主存时已经完成逻辑地址到物理地址的转换,在程序的执行期间将不会发生变化。其优点是:无须硬件地址转换机构的支持,只要求程序本身是可重定位的,它只对那些要修改的地址部分具有某种标识,由专门设计的程序来完成。



(2) 动态地址重定位,是指在程序运行期间完成逻辑地址到物理地址的转换。其实现机制要依赖硬件地址转换机构,如基址寄存器 BR。其优点是:程序在执行期间可以被换入和换出主存,以解决主存紧张的问题;可以在主存中移动,把主存中的碎片集中起来,以充分利用空间;不必给程序分配连续的主存空间,以便较好地利用较小的主存块,可以实现共享。

3. 存储器管理的功能

存储器管理的功能如下。

- 主存储器的分配和回收。
- 提高主存储器的利用率:减少碎片(也称零头),允许多道程序动态共享主存。
- 存储保护:任务是确保每道程序都在自己的主存空间运行,互不干扰。
- 主存扩充:主存扩充的任务是从逻辑上扩充主存容量,使用户认为系统所拥有的主存空间远比其实际的主存空间(RAM)大得多。

二、分区存储管理

存储管理主要包括分区存储管理、分页存储管理、分段存储管理、段页式存储管理和虚拟存储管理。其中分区存储管理是把主存的用户区划分成若干个区域,每个区域分配给一个用户作业使用,并限定它们只能在自己的区域中运行。按划分方式的不同,它可分为固定分区、可变分区和可重定位分区。

- 固定分区。它是一种静态分区方式,在系统生成时已将主存划分为若干个分区,每个分区的大小可不等。
- 可变分区。它是一种动态分区方式,存储空间的划分是在作业装入时进行的,故分区的个数可变,分区的大小刚好等于作业的大小。
- 可重定位分区。它是解决碎片问题简单而又行之有效的方法。其基本思想是,移动所有已分配好的分区,使之成为连续区域。

分区划分完成后的问题就是如何进行分区的保护,通常采用上界和下界寄存器保护法以及基址和限长寄存器保护法。

采用上界和下界寄存器保护法时,上界寄存器中存放作业的装入地址,下界寄存器装入作业的结束地址。形成的物理地址必须满足:

$$\text{上界寄存器} \leq \text{物理地址} \leq \text{下界寄存器}$$

采用基址和限长寄存器保护法时,基址寄存器中存放作业的装入地址,限长寄存器中装入作业的长度,形成的物理地址必须满足:

$$\text{基址寄存器} \leq \text{物理地址} \leq \text{基址寄存器} + \text{限长寄存器}$$

三、分页存储管理

1. 纯分页存储管理

分页原理:将一个进程的地址空间划分为若干个大小相等的区域,称为页。相应的,将内存空间划分成与页相同大小的若干个物理块,称为块或页框。

地址机构:分页系统的地址机构如图 2-2 所示,由两部分组成,页号 P 和偏移量 W(即页内地址)。图中的地址长度为 32 位,其中 0~11 位为页内地址(每页大小为 4KB),12~31



位为页号, 所以允许的地址空间大小最多为 1M 个页。

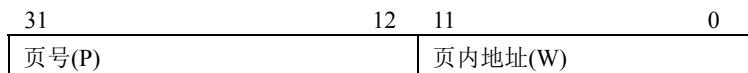


图 2-2 分页系统的地址机构

系统将用户程序的逻辑空间按照同样大小也划分成若干页面, 称为逻辑页面, 有时也简称为页。程序的各个逻辑页面从 0 开始依次编号, 称为逻辑页号或相对页号。每个逻辑页面内也从 0 开始编址, 称为页内地址。用户程序的逻辑地址由逻辑页号和页内地址两部分组成。

页表: 系统为每个进程建立一张页面映射表, 简称页表。页表用于记录用户程序逻辑页面与内存物理页面之间的对应关系。页表的作用是实现从页号到物理块号的地址映射。

地址变换机构: 其任务是利用页表把逻辑地址变换成内存中的物理地址。

2. 快表

在地址映射过程中, 共需两次访问内存。第一次是访问页表, 得到数据的物理地址, 第二次才是存取数据。显然, 这样就增加了访问的时间。在地址映射机制中增加一个小容量的联想寄存器(相联存储器), 它由高速寄存器组成一张快表, 用来存放当前访问最频繁的少数活动页的页号及相关信息。

快表只存放当前进程最活跃的少数几页, 随着进程的推进, 快表内容动态更新。当某一用户程序需要存取数据时, 根据该数据所在的逻辑页号在快表中找出对应的物理页号, 然后拼接页内地址, 以形成物理地址; 如果在快表中没有相应的逻辑页号, 则地址映射仍然通过内存中的页表进行。

四、分段存储管理

1. 基本原理

在分段存储管理方式中, 作业的地址空间被划分为若干个段, 每个段是一组完整的逻辑信息, 如有主程序段、子程序段、数据段及堆栈段等, 每个段都有自己的名字, 都是从 0 开始编址的一段连续的地址空间, 各段长度是不等的。

分段系统的逻辑地址由段号(名)和段内地址两部分组成。在该地址结构中, 允许一个作业最多有 64K 段, 每个段的最大长度为 64KB。

在分段式存储管理系统中, 为每个段分配一个连续的分区, 而进程中的各个段可以离散地分配到内存中不同的分区中。在系统中为每个进程建立一张段映射表, 简称为“段表”。每个段在表中占用一个表项, 在其中记录了该段在内存中的起始地址(又称为“基址”)和段的长度。进程在执行中, 通过查段表来找到每个段所对应的内存区。所以说, 段表实现了从逻辑段到物理内存区的映射。

2. 段的动态链接和装配

所谓动态链接, 是指在一个程序开始运行时, 只将主程序装配好并调入内存, 在运行过程中若访问一个新的模块时, 再装配此模块, 并与主程序链接起来。所以, 动态链接是以段为基础的。



在可变分区分配方案中,主存中放置的程序常采用首次适应、最佳适应或最差适应算法实现,但运行的程序需连续存放在一个分区中,一个作业是由若干个具有逻辑意义的段(如主程序、子程序、数据段等)组成的。分段系统中,允许程序(作业)占据主存中若干分离的分区。每个分区存储一个程序分段。这样,每个作业需要几对界限地址,判定访问地址是否越界就困难了。在分段存储系统中常常利用存储保护键来实现存储保护。

五、虚拟存储器管理

1. 虚拟存储器的引入

1) 局部性原理

存储管理策略的基础是局部性原理——进程往往会不均匀地、高度局部化地访问主存。局部性表现为时间局部性和空间局部性两类。

- 时间局部性:是指最近被访问的存储位置,很可能不久的将来还要访问,如循环、栈等。
- 空间局部性:是指存储访问有成组的倾向,当访问了某个位置后,很可能还要访问其附近的位置,如访问数组、代码顺序执行等。

2) 虚拟存储器的定义

根据局部性原理,一个作业在运行之前,没有必要全部装入主存,而仅将当前要运行的那部分页面或段先装入主存启动运行,其余部分暂时留在磁盘上。

程序在运行时如果所要访问的页(段)已调入主存,便可继续执行下去;但如果所要访问的页(段)尚未调入主存(称为缺页或缺段),程序应利用操作系统所提供的请求调页(段)功能,将它们调入主存,以使进程能继续执行下去。如果此时主存已满,无法再装入新的页(段),则还要再利用页(段)的置换功能,将主存中暂时不用的页(段)调出至磁盘上,以便腾出足够的主存空间后,再将所要访问的页(段)调入主存,使程序继续执行下去。这样,便可使一个大的用户程序在较小的主存空间中运行,也可使主存中同时装入更多的进程并发执行。从用户角度看,该系统所具有的主存容量,将比实际主存容量大得多,人们把这样的存储器称为虚拟存储器。

虚拟存储器具有请求调入功能和置换功能,能仅把作业的一部分装入主存便可运行作业的存储器系统,能从逻辑上对主存容量进行扩充。

3) 虚拟存储器的实现

请求分页系统:在分页系统的基础上,增加了请求调页功能和页面置换功能所形成的页式虚拟存储系统。

请求分段系统:在分段系统的基础上,增加了请求调段功能和分段置换功能所形成的段式虚拟存储系统。

2. 请求分页中的硬件支持

请求分页是目前常用的一种虚拟存储器方式。

1) 请求分页的页表机制

请求分页的页表机制是在纯分页的页表机制上形成的,由于只将应用程序的一部分调入主存,但还有一部分仍在磁盘上,故需在页表中再增加若干项,如状态位、访问字段、辅存地址等供程序(数据)在换进、换出时引用。





2) 缺页中断机构

在请求分页系统中,每当所要访问的页面不在主存时,便要产生一个缺页中断,请求操作系统将所缺页调入主存。它与一般中断的主要区别在于:缺页中断在指令执行期间产生和处理中断信号,而一般中断在一条指令执行完后检查和处理中断信号;缺页中断返回到该指令的开始重新执行该指令,而一般中断则返回到该指令的下一条指令执行。

3) 地址转换机构

请求分页系统中的地址转换机构是在分页系统的地址转换机构的基础上,为实现虚拟存储器而增加了某些功能后形成的,如产生和处理缺页中断、从主存中换出一页等。

3. 页面置换算法

1) 最佳置换算法

最佳(Optimal)置换算法是一种理想化的算法,性能最好,但实际上难以实现,所以该算法通常用来评价其他算法。

2) 先进先出置换算法

先进先出(FIFO)置换算法总是淘汰最先进入内存的页面。其算法实现简单,是一种最直观,也是性能最差的算法。

3) 最近最久未使用置换算法

最近最久未使用(Least Recently Used, LRU)置换算法是选择最近最久未使用的页面予以淘汰,系统在每个页面设置一个访问字段,用以记录这个页面自上次被访问以来所经历的时间 T,当要淘汰一个页面时,选择 T 最大的页面。

4) 最近未用置换算法

最近未用(Not Used Recently, NUR)置换算法将最近一段时间未引用过的页面换出,是一种 LRU 的近似算法。

2.2.4 设备管理

一、设备管理概述

1. 设备的分类

1) 按设备上数据组织方式分类

- 块设备:指以数据块为单位组织和传送数据的设备,如磁盘、磁带等。
- 字符设备:指以单个字符为单位存取信息的设备,如终端、打印机等。

2) 按资源分配的角度分类

- 独占设备:对这类设备来说,在一段时间内最多只能有一个进程占有并使用它。低速 I/O 设备一般是独占设备,如打印机、终端等。
- 共享设备:这类设备允许多个进程共享,即多个进程的 I/O 传输可以交叉。
- 虚拟设备:在一类设备上模拟另一类设备的技术称为虚设备技术。通常是用高速设备来模拟低速设备,以此把原来慢速的独占设备改造成能为若干进程共享的快速共享设备。就好像把一台设备变成了多台虚拟设备,从而提高了设备的利用率。我们称被模拟的设备为虚设备。Spooling 技术就是一类典型的虚设备技术。



3) 按数据传输率分类

- 低速设备：指传输速率为每秒钟几个字节到数百个字节的设备。典型的设备有键盘、鼠标、语音的输入等。
- 中速设备：指传输速率在每秒钟数千字节至数万字节的设备。典型的设备有行式打印机、激光打印机等。
- 高速设备：指传输速率在数十万个字节至数兆字节的设备。典型的设备有磁带机、磁盘机、光盘机等。

4) 其他分类方法

按输入/输出对象的不同，设备可分为人机通信、机机通信设备。

按是否可交互，设备可分为非交互设备，如机机通信设备、外存、卡带机等；交互设备，如终端等。

2. 设备管理的目标和任务

1) 操作系统设备管理的目标

- 向用户提供使用外围设备的方便、统一的接口。
- 充分利用中断技术、通道技术和缓冲技术，提高 CPU 与设备、设备与设备之间的并行工作能力，提高外围设备的使用效率。
- 保证在多道程序环境下，当多个进程竞争使用设备时，按照一定的策略分配和管理设备，以使系统能有条不紊地工作。

2) 设备管理的任务

设备管理的任务是保证在多道程序环境下，当多个进程竞争使用设备时，按一定策略分配和管理各种设备，控制设备的各种操作，完成输入/输出设备与内存之间的数据交换。

设备管理的主要功能如下所述。

- 动态地掌握并记录设备的状态。
- 设备分配和释放。
- 缓冲区管理。
- 实现物理输入/输出设备的操作。
- 提供设备使用的命令接口和编程接口。
- 设备的访问和控制，包括并发访问和差错处理。
- 输入/输出缓冲和调度，目的是提高输入/输出的访问效率。

二、DMA 与缓冲技术

1. DMA 技术

DMA 技术基本思想是：在外围设备和主存之间开辟直接的数据交换通路。在内存与输入/输出设备间传送一个数据块的过程中，不需要 CPU 的任何干涉，只需要 CPU 在过程开始启动与过程结束时的处理，实际操作由 DMA 硬件直接执行完成。

2. 缓冲技术

缓冲是计算机系统中常用的技术。一般来说，凡是数据到达速度和离去速度不匹配的地方都可以采用缓冲技术。缓冲可以采用硬件缓冲和软件缓冲两种技术。硬件缓冲是利用





专门的硬件寄存器作为缓冲区；软件缓冲是利用操作系统的管理，用主存中的一个或多个区域作为缓冲区，进而可以形成缓冲池。

3. Spooling 技术

Spooling 是外围设备联机操作(Simultaneous Peripheral Operations on Line)的缩写，常简称为 Spooling 系统或假脱机系统。假脱机技术实际上是用一类物理设备模拟另一类物理设备的技术，可以将低速的独占设备改造成一种可共享的设备，而且一台物理设备可以对应若干台虚拟的同类设备。Spooling 系统由“预输入程序”“缓输出程序”和“井管理程序”以及“输入和输出井”组成。

Spooling 系统将一个作业从进入系统到完成后撤离系统的全过程，划分成输入、处理和输出三个并发执行的过程。当用户作业要进入系统时，由 Spooling 系统的预输入程序将作业信息从物理输入设备上送到磁盘上的指定区域(称为输入井)。输入井中的作业有四种状态。

- 输入状态。作业的信息正从输入设备上预输入。
- 收容状态。作业预输入结束但未被选中执行。
- 执行状态。作业已被选中并处于运行过程中，它可从输入井中读取数据信息，也可向输出井写信息。
- 完成状态。作业已经撤离，该作业的执行结果等待缓输出。

Spooling 系统的引入缓和了 CPU 与设备速度的不均匀性，提高了 CPU 与设备的并行程度。

三、磁盘调度

对磁盘的存取访问一般要有三部分时间。首先要将磁头移动到相应的磁道或柱面上，这个时间叫作寻道时间；一旦磁头到达指定磁道，必须等待所需要的扇区旋转到读/写头下，这个时间叫作旋转延迟时间；信息在磁盘和内存之间的实际传送时间叫作传送时间。磁盘调度的目的是使平均寻道时间最少。

1. 磁盘驱动调度

一般可采用以下四种磁盘优化调度算法。

1) 先来先服务算法

先来先服务算法(FCFS)即按照访问请求的次序为各个进程服务，这是最公平且最简单的算法，但是效率不高。

2) 最短寻道时间优先算法

最短寻道时间优先算法(SSTF)以寻道优化为出发点，优先为距离磁头当前所在位置最近的磁道(柱面)的访问请求服务。这种算法使每次的寻道时间最短，但也存在缺点：不能保证平均寻道时间最短。

3) 扫描算法

扫描算法(SCAN)也是一种寻道优化的算法，它克服了 SSTF 算法的缺点，既考虑访问磁道与磁头当前位置的距离，又考虑磁臂的移动方向，且以方向优先。这种算法比较公平，而且效率较高。这种算法因其基本思想与电梯的工作原理相似，故又称为电梯算法。



4) 单向扫描调度算法

单向扫描调度算法(CSCAN)存在这样的问题：当磁头刚从里向外移动过某一磁道时，恰有一进程请求访问此磁道，这时该进程必须等待磁头从里向外，然后再从外向里扫描完所有要访问的磁道后，才处理该进程的请求，致使该进程的请求被严重地推迟。为了减少这种延迟，CSCAN 算法规定磁头作单向移动。

2. 旋转调度算法

系统应该选择延迟时间最短的进程对磁盘的扇区进行访问。当有若干等待进程请求访问磁盘上的信息时，旋转调度应考虑以下三种情况。

- 进程请求访问的是同一磁道上的不同编号的扇区。
- 进程请求访问的是不同磁道上的不同编号的扇区。
- 进程请求访问的是不同磁道上具有相同编号的扇区。



2.2.5 文件管理

一、文件与文件系统

1. 文件

文件是具有符号名的，在逻辑上具有完整意义的一组相关信息项的集合。文件可以有格式的，也可以是无格式的。

信息项是构成文件的基本单位，可以是一个字符，也可以是一个记录。一个文件包括文件体和文件说明。

2. 文件系统

操作系统的文件系统包括两个方面：一方面包括负责管理文件的一组系统软件；另一方面包括被管理的对象——文件。

文件系统的主要功能包括：按名存取、统一的用户接口、并发访问和控制、安全性控制、差错恢复等。

3. 文件类型

根据文件的性质和用途不同，文件有多种分类方法。

- 按文件的用途，文件可以分为系统文件、库文件和用户文件等。
- 按信息保存期限，文件可以分为临时文件、档案文件和永久文件。
- UNIX 系统将文件分为普通文件、目录文件和设备文件(特殊文件)等。
- 按文件的保护方式，文件可以分为只读文件、读写文件、可执行文件和不保护文件等。

目前常用的文件系统类型有：FAT、VFAT、NTFS、Ext2、HPFS 等。

文件分类的目的是对不同文件进行管理，提高系统效率和用户界面的友好性。

二、文件的结构和组织

文件的结构是指文件的组织形式，从用户角度所看到的文件组织形式，称为文件的逻辑结构；从实现角度考察文件在辅助存储器上的存放方式，常称为文件的物理结构。





1. 文件的逻辑结构

一般文件的逻辑结构可以分为两种：无结构的字符流文件和有结构的记录文件，后者也称为有格式文件。

记录文件可分为定长和不定长两类。

- 定长记录文件：指文件中所有记录的长度相同。
- 不定长记录文件：指文件中各记录的长度不相同。

在 UNIX 系统中，所有的文件都被看作流式文件，系统不对文件格式进行处理。

2. 文件的物理结构

文件的物理结构是指文件的内部组织形式，也就是文件在物理存储设备上的存放方法。常用的文件物理结构有以下三种。

1) 顺序结构

顺序结构又称连续结构。这是一种最简单的物理结构，它把逻辑上连续的文件信息依次存放在连续编号的物理块中。只要知道文件在存储设备上的起始地址(首块号)和文件长度(总块数)，就能很快地进行存取。这种结构的优点是访问速度快，缺点是很难增加文件的长度。

2) 链接结构

链接结构将逻辑上连续的文件分散存放在若干不连续的物理块中，每个物理块设有一个指针，指向其后续的物理块。只要指明文件的第一个块号，就可以利用链指针检索整个文件。这种结构的优点是文件长度容易动态变化，缺点是不适合随机存取访问。

3) 索引结构

采用索引结构时，系统为每个文件建立一张索引表，索引表中每一表项指出文件信息所在的逻辑块号和与之对应的物理块号。

对一些大的文件，索引表的大小超过一个物理块时，就会发生索引表的分配问题，一般采用多级(间接索引)技术。这时由索引表指出的物理块中存放的不是文件信息，而是存放文件信息的物理块地址。这样，如果一个物理块能存储 n 个物理块地址，则一次间接索引，可寻址的文件长度将变成 $n \times n$ 块。

UNIX 文件系统采用三级索引结构，文件系统中的 **inode** 是基本的构件，它表示文件系统树形结构的节点。UNIX 文件索引表项分为四种寻址方式：直接寻址、一级间接寻址、二级间接寻址和三级间接寻址。

三、文件目录

系统为每个文件设置一个描述性数据结构——文件控制块 FCB(File Control Block)，文件目录就是文件控制块的有序集合。

1. 文件控制块

文件控制块是系统为管理文件而设置的一个数据结构。FCB 是文件存在的标志，通常包含三类信息：基本信息类、存取控制信息类和使用信息类。

- 基本信息类：文件名、文件的物理位置、文件长度和文件块数等。
- 存取控制信息类：文件的存取权限。在 UNIX 系统中，将用户分成文件主、同组



用户和一般用户三类，它们具有不同的操作权限。

- 使用信息类：建立文件的日期和时间、最后访问日期和时间、最后修改日期和当前使用的信息等。

为了实现文件目录的管理，通常将文件目录以文件的形式保存在外存空间，这个文件就被称为目录文件。目录文件是长度固定的记录式文件。

2. 目录结构

文件目录的组织与管理是文件管理中的一个重要方面，常见的目录结构有三种：一级目录结构、二级目录结构和多级目录结构。

多级目录结构像一棵倒置的有根树，所以也称为树形目录结构。从树根向下，每一个节点是一个目录，叶节点是文件。DOS 和 UNIX 等操作系统均采用多级目录结构。

在采用多级目录结构的文件系统中，用户要访问一个文件，必须指出文件所在的路径名。路径名包含从根目录开始到该文件的通路上所有各级目录名。各级目录名之间，目录名与文件名之间需要用分隔符隔开。例如，在 DOS 中分隔符为“\”，在 UNIX 中分隔符为“/”。绝对路径名是指从根目录开始的完整文件名，即由从根目录开始的所有目录名以及文件名构成的。

四、存取方法和存取控制

1. 文件的存取方法

文件的存取方法是指读写文件存储器上的一个物理块的方法，通常有顺序存取、随机存取和按键存取等方法。

1) 顺序存取

顺序存取就是按从前到后的次序依次访问文件的各个信息项。对于记录式文件，是按记录的排列顺序来存取的。对流式文件，顺序存取反映当前读写指针的变化，在存取完一段信息后，读写指针自动指出下次存取时的位置。

2) 随机存取

随机存取又称直接存取，即允许用户根据记录键存取文件的任意记录，或者是根据存取命令把读写指针移到指定处读写。

3) 按键存取

按键存取法是直接存取法的一种，它不是根据记录的编号或地址来存取文件中的记录，而是根据文件中各记录的某个数据项内容来存取记录的，这种数据项称之为“键”。因此，将这种存取法称为按键存取法。

2. 文件存储空间的管理

外存空间管理的数据结构通常称为磁盘分配表(Disk Allocation Table)。常用的空间管理方法有位示图、空闲区表、空闲块链和成组链接法。

1) 位示图

在外存上建立一张位示图(Bitmap)，记录文件存储器的使用情况。每一位对应文件存储器上的一个物理块，取值 0 和 1 分别表示空闲和占用。文件存储器上的物理块依次编号为 0, 1, 2, …, 假如系统中字长为 32 位，那么在位示图中的第一个字对应文件存储器上的 0, 1,





2, ..., 31 号物理块; 第二个字对应文件存储器上的 32, 33, 34, ..., 63 号物理块; 依次类推。这种方法的主要特点是位示图的大小由磁盘空间的大小(物理块总数)决定, 位示图的描述能力强, 适用于各种物理结构。

2) 空闲区表

将外存空间上一个连续未分配区域称为“空闲区”。操作系统为磁盘外存上所有空闲区建立一张空闲表, 每个表项对应一个空闲区, 空闲表中包含序号、空闲区的第一块号、空闲块的块数等信息。它适用于连续文件结构。

3) 空闲块链

每个空闲物理块中都有指向下一个空闲物理块的指针, 所有空闲物理块构成一个链表, 链表的头指针放在文件存储器的特定位置上(如管理块中)。

4) 成组链接法

在 UNIX 系统中, 将空闲块分成若干组, 每 100 个空闲块为一组, 每组的第一个空闲块登记了下一组空闲块的物理盘块号和空闲块总数, 假如一个组的第一个空闲块号等于 0 的话, 意味着该组是最后一组, 即无下一组空闲块。

五、文件的使用

操作系统在操作级(命令级)和编程级(系统调用和函数)向用户提供文件的服务。操作系统在操作级向用户提供的命令有目录管理类命令、文件操作类命令(如复制、删除和修改)、文件管理类命令(如设置文件权限)等。操作系统在编程级向用户提供的系统调用主要有以下六种。

- 创建文件: 如 `create`(文件名, 参数表)。
- 删除文件: 如 `delete`(文件名)。
- 打开文件: 如 `open`(文件名, 参数表)。
- 关闭文件: 如 `close`(文件名)。
- 读文件: 如 `read`(文件名, 参数表)。
- 写文件: 如 `write`(文件名, 参数表)。

六、文件的共享和保护

1. 文件的共享

文件共享是指不同用户进程使用同一文件。常见的文件链接有硬链接和符号链接两种。

1) 硬链接

文件的硬链接是指两个文件目录表目指向同一个索引节点的链接, 该链接也称基于索引节点的链接。文件硬链接不利于文件主要删除它所拥有的文件, 因为文件主要删除它所拥有的共享文件, 必须首先删除(关闭)所有的硬链接, 否则就会造成共享该文件的用户目录表目指针悬空。

2) 符号链接

符号链接是指建立的新的文件或目录与原来文件或目录的路径名映射。当访问一个符号链接时, 系统通过该映射找到原文件的路径, 并对其进行访问。符号链接的缺点: 其他用户读取符号链接的共享文件和读取硬链接的共享文件相比, 需要增加读盘操作的次数。



2. 文件的保护

文件系统对文件的保护常采用存取控制方式进行, 所谓存取控制, 就是不同的用户对文件的访问有不同的权限, 以防止文件被未经文件主同意的用户访问。

1) 存取控制矩阵

理论上存取控制方法可采用存取控制矩阵实现, 它是一个二维矩阵, 一维列出计算机的全部用户, 另一维列出系统中的全部文件, 矩阵中每个元素 A_{ij} 是表示第 i 个用户对第 j 个文件的存取权限。通常存取权限有可读、可写、可执行以及它们的组合。

2) 存取控制表

存取控制表是按用户对文件的访问权限的差别对用户进行分类, 由于某一文件往往只与少数几个用户有关, 所以这种分类方法可使存取控制表大为简化。UNIX 系统使用的就是这种存取控制表方法。它把用户分成三类, 包括文件主、同组用户和其他用户, 每类用户的存取权限为可读、可写、可执行的组合。

3) 用户权限表

用户权限表是以用户或用户组为单位将用户可存取的文件集中起来存入表中, 表中每个表目表示该用户对相应文件的存取权限, 这相当于存取控制矩阵一行的简化。

4) 密码

在创建文件时, 由用户提供一个密码, 在文件存入磁盘时用该密码对文件内容进行加密。进行读取操作时, 要对文件进行解密, 只有知道密码的用户才能读取文件。

七、系统的安全与可靠性

1. 系统的安全

系统的安全涉及两类问题: 一类涉及技术、管理、法律、道德和政治等问题; 另一类涉及操作系统的安全机制。

一般从四个级别上对文件进行安全性管理: 系统级、用户级、目录级和文件级。

文件级安全管理是通过系统管理员或文件主对文件属性的设置来控制用户对文件的访问。通常可设置的属性有: 只执行、隐含、只读、读写、共享、系统。用户对文件的访问由用户访问权、目录访问权和文件属性三者的权限决定。

2. 文件系统的可靠性

文件系统的可靠性是指系统抵抗和预防各种物理性破坏和人为性破坏的能力。

1) 转储和恢复

常用的转储方法有: 静态转储和动态转储、海量转储和增量转储。

2) 日志文件

在计算机系统工作的过程中, 操作系统把用户对文件的插入、删除和修改的操作写入日志文件。一旦发生故障, 可利用日志文件来进行系统故障的恢复。

3) 文件系统的一致性

影响文件系统可靠性的因素之一是文件系统的一致性问题。通常解决方案是采用文件系统的一致性检查, 包括块的一致性检查和文件的一致性检查。



2.2.6 作业管理

一、作业管理概述

作业是系统为完成一个用户的计算任务(或一次事务处理)所做的工作总和。

1. 作业控制

用户可以采用脱机和联机两种控制方式来控制作业的运行。在脱机控制方式中,作业运行的过程是无须人工干预的,因此用户必须编写成作业说明书连同作业一起提交给计算机系统。在联机控制方式中,操作系统向用户提供了一组联机命令,用户可以通过终端输入命令,将自己的意图告诉计算机,以控制作业的运行过程。

作业由程序、数据和作业说明书三部分组成。作业说明书包括作业基本情况、作业控制、作业资源要求的描述。其中:作业基本情况包括用户名、作业名、编程语言、最大处理时间等;作业控制描述包括作业控制方式、作业步的操作顺序、作业执行的出错处理;作业资源要求描述包括处理时间、优先级、主存空间、外设类型和数量、实用程序要求等。

2. 作业状态

作业的状态分为四种:提交、后备、执行和完成。

- 提交:作业提交给计算机中心,通过输入设备送入计算机系统的过程所处的状态称之为提交状态。
- 后备:作业通过 Spooling 系统输入计算机系统的后备存储器(磁盘)中,随时等待作业调度程序调度时的状态。
- 执行:一旦作业被作业调度程序选中,为其分配了必要的资源,并为其建立相应的进程后,该作业便进入了执行状态。
- 完成:当作业正常结束或异常终止时,便进入完成状态。此时由作业调度程序对该作业进行善后处理。如撤销作业的作业控制块,收回作业所占的系统资源,将作业的执行结果形成输出文件放到输出井中,由 Spooling 系统控制输出。

3. 作业控制块(JCB)和作业后备队列

作业控制块是指记录与该作业有关的各种信息的登记表。作业控制块是作业存在的唯一标志。通常将作业控制块排成一个或多个队列,而这些队列通常称为作业后备队列,即作业后备队列是由若干个作业控制块组成的。

二、作业调度

作业调度主要是从后备状态的作业中挑选一个(或一些)作业投入运行。根据不同的调度目标,有不同的调度算法。

1. 调度算法的选择

调度算法的选择主要有以下五点原则:响应时间快;周转时间或加权周转时间短;均衡的资源利用率;吞吐量大;系统反应时间短。

2. 调度算法

作业调度算法有许多种,常见的有如下几种。



- 先来先服务(FCFS): 按作业到达的先后次序调度, 它不利于短作业。
- 短作业优先(SJF): 按作业的估计运行时间调度, 估计运行时间短的作业优先调度, 它不利于长作业, 可能会使一个估计运行时间长的作业迟迟得不到服务。
- 响应比高者优先(HRN): 综合上述两者, 既考虑作业估计运行时间, 又考虑作业等待时间, 响应比是:

$$\text{HRN} = (\text{估计运行时间} + \text{等待时间}) / \text{估计运行时间}$$

- 优先级调度算法: 根据作业的优先级别, 优先级高者先调度。
- 均衡调度算法: 根据系统运行情况和作业本身进行分类, 调度程序从这些分类中轮流挑选作业执行。

3. 作业调度算法性能的衡量指标

在一个以批量处理为主的系统中, 通常用平均周转时间或平均周转系数来衡量调度性能的优劣。假设作业 $J_i (i=1, 2, \dots, n)$ 的提交时间为 t_{si} , 执行时间为 t_{ri} , 作业完成时间为 t_{oi} , 则作业 J_i 的周转时间 T_i 和周转系数 W_i 分别定义为:

$$T_i = t_{oi} - t_{si} \quad (i=1, 2, \dots, n)$$

$$W_i = T_i / t_{ri} \quad (i=1, 2, \dots, n)$$

n 个作业的平均周转时间 r 和平均周转系数 W 分别定义为:

$$T = \frac{1}{n} \sum_{i=1}^n T_i, \quad W = \frac{1}{n} \sum_{i=1}^n W_i$$

从用户的角度来说, 总是希望自己的作业在提交后能立即执行, 这就意味着当等待时间为 0 时作业的周转时间最短, 即 $T_i = t_{ri}$ 。但是作业的执行时间 t_{ri} 并不能直观地衡量出系统的性能, 而周转系数 W_i 却能直观地反映系统的调度性能。从整个系统的角度来说, 不可能满足每个用户的这种要求, 而只能是系统的平均周转时间或平均周转系数最小。

三、人机界面

人机界面是计算机中实现用户与计算机通信的软件和硬件部分的总称。人机界面也称为用户接口或用户界面。从计算机用户界面的发展过程来看, 人机界面的发展可分为以下四个阶段。

- (1) 控制面板式用户界面。
- (2) 字符用户界面。
- (3) 图形用户界面。
- (4) 新一代用户界面。

2.3 真题详解

综合知识试题

试题 1 (2017 年下半年试题 23)

计算机加电自检后, 引导程序首先装入的是 (23), 否则, 计算机不能做任何事情。





- (23) A. Office 系列软件 B. 应用软件
C. 操作系统 D. 编译程序

参考答案: (23) C

要点解析: 操作系统是在硬件之上, 所有其他软件之下, 是其他软件的共同环境与平台。操作系统的主要部分是被频繁用到的, 因此是常驻内存的(Reside)。计算机加电以后, 首先引导操作系统。不引导操作系统, 计算机不能做任何事。

试题 2 (2017 年下半年试题 25)

当一个双处理器的计算机系统中同时存在 3 个并发进程时, 同一时刻允许占用处理器的进程数 (25)。

- (25) A. 至少为 2 个 B. 最多为 2 个
C. 至少为 3 个 D. 最多为 3 个

参考答案: (25) B

要点解析: 一个双处理器的计算机系统中尽管同时存在 3 个并发进程, 但同一时刻允许占用处理器的进程数最多为 2 个。

试题 3 (2017 年下半年试题 26)

假设系统有 $n(n>5)$ 个并发进程共享资源 R, 且资源 R 的可用数为 2。若采用 PV 操作, 则相应的信号量是 S 的取值范围应为 (26)。

- (26) A. $-1 \sim n-1$ B. $-5 \sim 2$ C. $-(n-1) \sim 1$ D. $-(n-2) \sim 2$

参考答案: (26) D

要点解析: 初始值资源数为 2, n 个并发进程申请资源, 信号量最大为 2, 最小为 $2-n$ 。

试题 4 (2017 年下半年试题 27)

在磁盘移臂调度算法中, (27) 算法在返程时不响应里道访问磁盘的请求。

- (27) A. 先来先服务 B. 电梯调度 C. 单向扫描 D. 最短寻道时间优先

参考答案: (27) C

要点解析: 在操作系统中常用的磁盘调度算法有: 先来先服务调度算法、最短寻道时间优先调度算法、电梯调度算法、单向扫描调度算法等。

移臂调度算法又叫磁盘调度算法, 根本目的在于有效利用磁盘、保证磁盘的快速访问。

① 先来先服务调度算法: 该算法实际上不考虑访问者要求访问的物理位置, 而只是考虑访问者提出访问请求的先后次序。有可能随时改变移动臂的方向。

② 最短寻道时间优先调度算法: 从等待的访问者中挑选寻道时间最短的那个请求执行, 而不管访问者的先后次序。这也有可能随时改变移动臂的方向。

③ 电梯调度算法: 从移动臂当前位置沿移动方向选择最近的那个柱面的访问者来执行, 若该方向上无请求访问时, 就改变臂的移动方向再选择。

④ 单向扫描调度算法。不考虑访问者等待的先后次序, 总是从 0 号柱面开始向里道扫描, 按照各自所要访问的柱面位置的次序去选择访问者。在移动臂到达最后一个柱面后, 立即快速返回到 0 号柱面, 返回时不为任何的访问者提供服务, 在返回到 0 号柱面后, 再次进行扫描。

试题5 (2017年上半年试题25)

在操作系统的进程管理中若系统中有6个进程要使用互斥资源R,但最多只允许2个进程进入互斥段(临界区),则信号量S的变化范围是__(25)___。

- (25) A. $-1 \sim 1$ B. $-2 \sim 1$ C. $-3 \sim 2$ D. $-4 \sim 2$

参考答案: (25) D

要点解析: 信号量初值为2,当有进程运行时,其他进程访问信号量,信号量就会减1,因此最小值为 $2-6=-4$ 。信号量S的变化范围为: $-4 \sim 2$ 。

2.4 强化训练

2.4.1 综合知识试题

试题1

在操作系统文件管理中,通常采用__(1)___来组织和管理外存中的信息。

- (1) A. 字处理程序 B. 设备驱动程序 C. 文件目录 D. 语言翻译程序

试题2和试题3

假设系统中进程的三态模型如图2-3所示,图中的a、b和c的状态分别为__(2)___;当运行进程执行P操作后,该进程__(3)___。

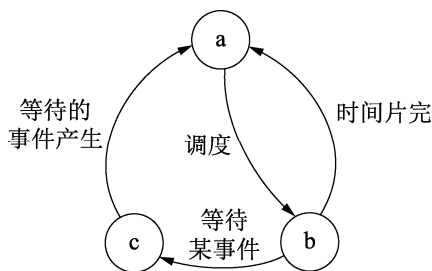


图 2-3 进程的三态模型

- (2) A. 就绪、运行、阻塞 B. 运行、阻塞、就绪
C. 就绪、阻塞、运行 D. 阻塞、就绪、运行
- (3) A. 处于运行状态 B. 处于阻塞状态
C. 处于就绪状态 D. 处于运行状态或者进入阻塞状态

试题4和试题5

在网络操作系统环境中,当用户A的文件或文件夹被共享时,__(4)___,这是因为访问用户A的计算机或网络的人__(5)___。

- (4) A. 其安全性与未共享时相比将会有所提高
B. 其安全性与未共享时相比将会有所下降
C. 其可靠性与未共享时相比将会有所提高



- D. 其方便性与未共享时相比将会有所下降
- (5) A. 只能够读取, 而不能修改共享文件夹中的文件
B. 可能能够读取, 但不能复制或更改共享文件夹中的文件
C. 可能能够读取、复制或更改共享文件夹中的文件
D. 不能够读取、复制或更改共享文件夹中的文件

试题 6

在磁盘移臂调度算法中, (6) 算法可能会随时改变移动臂的运动方向。

- (6) A. 电梯调度和先来先服务 B. 先来先服务和单向扫描
C. 电梯调度和最短寻道时间优先 D. 先来先服务和最短寻道时间优先

试题 7 和试题 8

在网络操作系统环境中, 当用户 A 的文件或文件夹被共享时, (7), 这是因为访问用户 A 的计算机或网络的人 (8)。

- (7) A. 其安全性与未共享时相比将会有所提高
B. 其安全性与未共享时相比将会有所下降
C. 其可靠性与未共享时相比将会有所提高
D. 其方便性与未共享时相比将会有所下降
- (8) A. 只能够读取, 而不能修改共享文件夹中的文件
B. 可能能够读取, 但不能复制或更改共享文件夹中的文件
C. 可能能够读取、复制或更改共享文件夹中的文件
D. 不能够读取、复制或更改共享文件夹中的文件

试题 9

某有限状态自动机的状态图如图 2-4 所示(状态 0 是初态, 状态 2 是终态), 则该自动机不能识别 (9)。

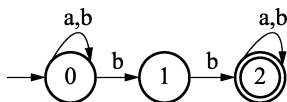


图 2-4 某有限状态自动机的状态图

- (9) A. abab B. aabb C. bbaa D. bbab

试题 10 和试题 11

假设某企业有一个仓库。该企业的生产部员工不断地将生产的产品送入仓库, 销售部员工不断地从仓库中取产品。假设该仓库能容纳 n 件产品。采用 PV 操作实现生产和销售的同步模型如图 2-5 所示, 该模型设置了三个信号量 S、S1 和 S2, 其中信号量 S 的初值为 1, 信号量 S1 的初值为 (10), 信号量 S2 的初值为 (11)。

- (10) A. -1 B. 0 C. 1 D. N
(11) A. -1 B. 0 C. 1 D. N

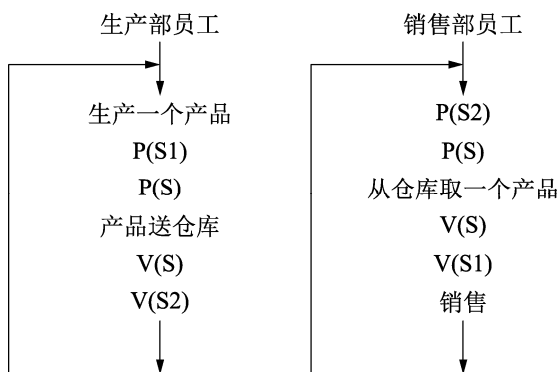


图 2-5 采用 PV 操作实现生产和销售的同步模型

2.4.2 综合知识试题参考答案

【试题 1】答案：(1) C

解析：操作系统中的文件系统专门负责管理外存储器上的信息，使用户可以“按名”高效、快速和方便地存储信息。为了实现“按名存取”，系统必须为每个文件设置用于描述和控制文件的数据结构，它至少要包括文件名和存放文件的物理地址，这个数据结构称为文件控制块，文件控制块的有序集合称为文件目录。换句话说，文件目录是由文件控制块组成的，专门用于文件检索。文件控制块也称为文件的说明或文件目录项(简称目录项)。

【试题 2 和试题 3】答案：(2)A (3)D

解析：试题(2)选 A。在多道程序系统中，进程的运行是走走停停，在处理器上交替运行，状态也不断地发生变化，因此进程一般有三种基本状态：运行、就绪和阻塞，也称为三态模型，如图 2-6 所示。

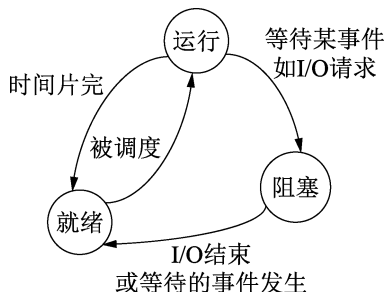


图 2-6 进程的三种基本状态

① 运行：当一个进程在处理器上运行时，称该进程处于运行状态。显然，对于单处理机系统，处于运行状态的进程只有一个。②就绪：一个进程获得了除处理器外的一切所需资源，一旦得到处理器即可运行，则称此进程处于就绪状态。③阻塞：也称等待或睡眠状态，一个进程正在等待某一事件发生(例如，请求 I/O 而等待 I/O 完成等)而暂时停止运行，这时即使把处理器分配给该进程，它也无法运行，故该进程处于阻塞状态。

试题(3)选 D。PV 操作是实现进程同步与互斥的常用方法。P 操作和 V 操作是低级通信原语，在执行期间不可分割。其中，P 操作表示申请一个资源，V 操作表示释放一个资源。



P 操作的定义: $S: S-1$, 若 $S \geq 0$, 则执行 P 操作的进程继续执行; 若 $S < 0$, 则置该进程为阻塞状态(因为无可用资源), 并将其插入阻塞队列。题中, 将现在运行进程执行 P 操作, 即将 b 进程执行 P 操作, 此时, 如若系统中还存在剩余资源空间, 则 b 进程继续处于运行状态, 若系统中没有剩余资源空间, 则 b 进程进入阻塞状态。

【试题 4 和试题 5】答案: (4) B (5) C

解析: 在操作系统中, 用户 A 可以共享存储在计算机、网络和 Web 上的文件和文件夹, 但当用户 A 共享文件或文件夹时, 其安全性与未共享时相比将会有所下降, 这是因为访问用户 A 的计算机或网络的人可能能够读取、复制或更改共享文件夹中的文件。

【试题 6】答案: (6) D

解析: 在操作系统中常用的磁盘调度算法有: 先来先服务、最短寻道时间优先、扫描算法、循环扫描算法等。其中, 先来先服务是最简单的磁盘调度算法, 它根据进程请求访问磁盘的先后次序进行调度, 所以该算法可能会随时改变移动臂的运动方向。最短寻道时间优先算法根据进程请求, 访问磁盘的寻道距离短的优先调度, 因此该算法可能会随时改变移动臂的运动方向。电梯调度法的工作原理是先响应同方向(向内道或向外道方向)的请求访问, 然后再响应反方向的请求访问, 如同电梯的工作原理一样, 因此该算法可能会随时改变移动臂的运动方向。单项扫描算法是电梯调度法的改进, 该算法在返程时不响应请求访问, 是为了解决电梯调度法带来的饥饿问题。

【试题 7 和试题 8】答案: (7) B (8) C

解析: 在操作系统中, 用户 A 可以共享存储在计算机、网络和 Web 上的文件和文件夹, 但当用户 A 共享文件或文件夹时, 其安全性与未共享时相比将会有所下降, 这是因为访问用户 A 的计算机或网络的人可能能够读取、复制或更改共享文件夹中的文件。

【试题 9】答案: (9) A

解析: 对于选项 A, 从状态图的状态 0 出发, 识别 a 后到达状态 0, 识别 b 后到达状态 1, 由于不存在从状态 1 出发识别 a 的状态转移, 因此, abab 不能被该自动机识别。对于选项 B, 识别 aabb 的状态转移路是状态 0→状态 0→状态 1→状态 2。对于选项 C, 识别 bbaa 的状态转移路是状态 0→状态 1→状态 2→状态 2→状态 2。对于选项 D, 识别 bbab 的状态转移路是状态 0→状态 1→状态 2→状态 2→状态 2。

【试题 10 和试题 11】答案: (10) D (11) B

解析: 由于仓库能容纳 n 个产品, 需要设置一个信号量 S1, 且初值为 n, 表示仓库有存放 n 个产品的空间, 可以将产品送入缓冲区。为了实现生产部员工与销售部员工间的同步问题, 设置另一个信号量 S2, 且初值为 0, 表示缓冲区是否有产品。这样, 当生产部员工将生产产品送入缓冲区时, 需要判断缓冲区是否为空, 需要执行 P(S1), 产品放入缓冲区后需要执行 V(S2), 通知销售部仓库已经有产品。而销售部员工在取产品销售之前必须判断仓库是否有产品, 需要执行 P(S2), 取走产品后仓库空出一个存储单元, 需要执行 V(S1)。