

# 第3章 操作系统基础知识

根据考试大纲，本章要求考生掌握以下知识点：

- (1) 操作系统的内核（中断控制）、进程、线程概念；
- (2) 处理机管理（状态转换、共享与互斥、分时轮转、抢占、死锁）；
- (3) 存储管理（主存保护、动态连接分配、分段、分页、虚存）；
- (4) 设备管理（I/O 控制、假脱机）；
- (5) 文件管理（文件目录、文件组织、存取方法、存取控制、恢复处理）；
- (6) 作业管理（作业调度、作业控制语言、多道程序设计）；
- (7) 汉字处理，多媒体处理，人机界面；
- (8) 网络操作系统和嵌入式操作系统基础知识；
- (9) 操作系统的配置。

## 3.1 操作系统的功能、类型和层次结构

操作系统的定义、功能、类型和层次结构虽然在历年试题中没有涉及，但这是理解操作系统的工作机制的基础，需要深入理解和掌握。重点理解操作系统的定义和功能。

### 1. 操作系统定义

任何一个计算机系统都是由两个部分组成的：计算机硬件系统和计算机软件系统。操作系统（Operating System, OS）是计算机系统中的核心系统软件，负责管理和控制计算机系统中硬件和软件资源，合理地组织计算机工作流程和有效利用资源，在计算机与用户之间起接口的作用，如图 3-1 所示。



图 3-1 操作系统与硬件/软件的关系

在计算机系统中引入操作系统的目的可以从四个方面来理解。

### (1) 用户观点

操作系统是用户与计算机之间的接口。一方面，用户可以透明地使用计算机软/硬件资源；另一方面，操作系统提供了一些功能强大的系统调用，用户软件可以使用这些系统调

用请求操作系统服务。

#### （2）资源管理观点

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

#### （3）进程观点

操作系统由一个系统核心和若干并发运行的程序组成。这些运行的程序称为“进程”，进程可以分为用户进程和系统进程两大类。每个进程完成特定的任务，系统核心则控制和协调这些进程的运行。

#### （4）分层观点

操作系统通常采用分层结构实现，各层次的程序按照一定的结构组织并协调工作。

### 2. 操作系统分类

操作系统的基本类型有：

- Ⅰ 批处理操作系统（Batch Processing Operating System）
- Ⅰ 分时操作系统（Time Share Operating System）
- Ⅰ 实时操作系统（Real Time Operating System）
- Ⅰ 网络操作系统（Network Operating System）
- Ⅰ 分布式操作系统（Distributed Operating System）

### 3. 操作系统的功能

从资源管理的观点看，操作系统的功能分成五大部分，即：处理机管理、存储管理、文件管理、设备管理和作业管理。这五大部分相互配合，协调工作，实现对计算机系统的资源管理和控制程序的执行，为用户提供方便的使用接口和良好的运行环境。

### 4. 操作系统的结构设计模式

#### （1）模块化结构

操作系统由许多标准的、可兼容的基本单位构成（称为模块），各模块相对独立，模块之间通过规定的接口相互调用。模块化设计方法的优点是缩短了系统的开发周期，缺点是模块之间调用关系复杂、相互依赖，从而使分析、移植和维护系统较易出错。

#### （2）层次化结构

层次化结构是指操作系统由若干模块按照某种逻辑关系进行分层组合而成，各层之间只能单向依赖。优点是大大地简化了接口的设计，整个系统的正确性由各层次的正确性来保证，易于保证可靠性，也便于维护和移植。

#### （3）客户/服务器结构

操作系统中只包括一个最小的核心，操作系统的其他功能（如文件服务、进程服务等）由用户进程来实现。优点是服务器以用户进程的形式运行而不是运行于核心态，它们不直接访问硬件，单个服务器的崩溃不会导致整个系统的崩溃，它适用于分布式系统。

#### （4）对象模式

面向对象的操作系统是按照面向对象思想设计的，具有数据隐藏及消息激活对象等特征。其中对象是对操作系统管理的信息和资源的抽象，可以被视为受保护的信息或资源的总称。优点是适用于网络操作系统和分布式操作系统中，Windows NT 就被称为对象操作系统。

### (5) 对称多处理模式

如果一个操作系统在系统中的所有处理机运行且共享同一内存，这样的系统就是一个对称多处理系统。优点是适合共享存储器结构的多处理机系统，即紧耦合的多处理机系统。

## 3.2 处理机管理（进程管理）

进程管理是操作系统部分的核心内容，也是历年的考试重点，从1991年到2003年，共有10题涉及进程管理的知识点，占操作系统总题量的50%。从历年的考查情况看，主要偏重于进程的同步与互斥、信号量和P-V操作、进程的基本概念、管程，以及线程等方面。考生对进程部分的知识点应全面掌握。

### 1. 进程的概念

进程是可以与其他程序并发执行的段程序的一次执行过程，是系统进行资源分配和调度的基本单位。进程是一个程序关于某个数据集的一次运行。也就是说，进程是运行中的程序，是程序的一次运行活动。相对于程序，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。

从静态的角度看，进程实体由程序块、进程控制块（简称PCB）和数据块三部分组成。程序块描述该进程所要完成的任务；数据块包括程序在执行时所需要的数据和工作区。进程控制块包括进程的描述信息、控制信息、资源管理信息和CPU现场保护信息等，反映了进程的动态特性，如图3-2所示。

进程标识	状态	优先级	控制信息	队列	访问权限	现场
------	----	-----	------	----	------	----

图3-2 进程控制块PCB

PCB是进程存在的惟一标志，PCB描述了进程的基本情况。系统根据PCB感知进程的存在和通过PCB中所包含的各项变量的变化，掌握进程所处的状态以达到控制进程活动的目的。在创建一个进程时，首先创建其PCB，然后才能根据PCB中的信息对进程实施有效的管理和控制。当一个进程完成其功能后，系统则释放PCB，进程也随之消亡。一般情况下，进程的PCB结构都是全部或部分常驻内存的。

### 2. 进程的状态转换与控制

#### (1) 进程的状态及其转化

- 就绪状态。指进程分配到除处理机以外的必需的资源（已经具备了执行的条件）的状态。进程被创建后处于就绪状态，处于就绪状态的进程可以有多个。
- 执行状态。指进程占有处理机正在CPU上执行的状态。在单CPU系统中，每一时刻只有一个进程处于执行状态。
- 阻塞状态。指进程因等待某个事件的发生而放弃处理机进入等待状态。系统中处于这种状态的进程可以有多个。

现代操作系统还有挂起状态，进程的状态及转换如图3-3所示。

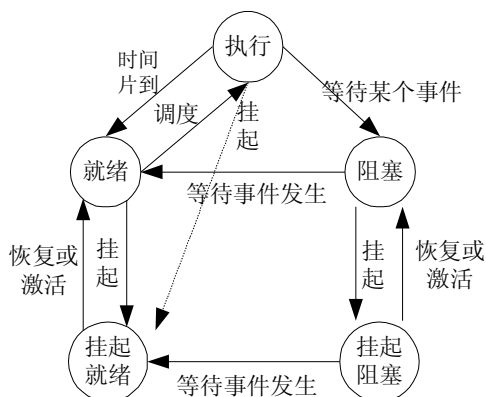


图 3-3 进程状态转换

进程的状态随着自身的推进和外界的变化而变化。比如，就绪状态的进程被进程调度程序选中进入执行状态；执行状态的进程因等待某一事件的发生转入等待状态；等待状态的进程在等待事件来到后便进入就绪状态。进程的状态可以动态地相互转换，但阻塞状态的进程不能直接进入执行状态，就绪状态的进程不能直接进入阻塞状态。在任何时刻，任何进程都处于且只能处于某一状态。

### （2）进程控制

进程控制是通过进程控制原语实现的。用于进程控制的原语主要包括，创建原语、阻塞原语、撤销原语、唤醒原语、优先级原语、调度原语。在操作系统中，原语是一个不可分割的基本单位。它们可以被系统本身调用，有的也以软中断形式供用户进程调用。

创建原语创建一个进程，包括系统创建和父进程创建都必须调用创建原语。新建的进程开始处于就绪状态。调度原语是按照确定的算法，从就绪队列中选择一个就绪进程，将处理器分配给它；修改这个进程的 PCB 内容。唤醒原语负责叫醒阻塞队列具备运行条件的某进程，使其回到就绪队列。撤销原语将执行完毕的进程登记、回收资源并撤销这个进程及其子进程。

通常操作系统中设置三种队列：执行队列、就绪队列和阻塞队列。在单处理器系统中执行队列只有一个成员。一般阻塞队列的个数取决于等待事件（原因）的个数。新创建的进程处于就绪队列。

## 3. 进程互斥与同步及 P、V 操作

### （1）进程互斥与同步的定义

进程互斥定义为，一组并发进程中一个或多个程序段，因共享某一公有资源而导致它们必须以一个不允许交叉执行的单位执行。也就是说，互斥是要保证临界资源在某一时刻只被一个进程访问。

进程同步定义为，异步环境下的一组并发进程因直接制约而互相发送消息，进行互相合作、互相等待，使得各进程按一定的速度执行的过程称为进程同步。也就是说，进程之间是异步执行的，同步即是使各进程按一定的制约顺序和速度执行。

### （2）信号量（Semaphore）与 P - V 操作

信号量可以有效地实现进程的同步和互斥。在操作系统中，信号量是一个整数。当信

号量大于等于零时，代表可供并发进程使用的资源实体数，当信号量小于零时表示正在等待使用临界区的进程数。建立一个信号量必须说明所建信号量所代表的意义和设置初值，以及建立相应的数据结构，以便指向那些等待使用该临界区的进程。

对信号量只能施加特殊的操作：**P**操作和**V**操作，**P**操作和**V**操作都是不可分割的原子操作，也称为原语，因此，**P - V**原语执行期间不允许中断发生。

**P (sem)**操作的作用是将信号量 **sem** 值减 1，若 **sem** 的值成负数，则调用 **P** 操作的进程暂停执行，直到另一个进程对同一信号量做 **V** 操作。**V(sem)** 操作的作用是将信号量 **sem** 值加 1，若 **sem** 的值小于等于 0，从相应队列（与 **sem** 有关的队列）中选择一个进程，唤醒它。

一般 **P** 操作与 **V** 操作的定义如下所述。

**P** 操作：

```
P(sem){
    sem = sem - 1;
    if(sem<0) 进程进入等待状态;
    else 继续进行; }
```

**V** 操作：

```
V(sem){
    sem = sem + 1;
    if(sem≤0) 唤醒队列中的一个等待进程;
    else 继续进行; }
```

### (3) 用 **P - V** 操作实现进程互斥

为了保护共享资源（如公共变量等），使它们不被多个进程同时访问，就要阻止这些进程同时执行访问这些资源的代码段，这些代码段称为临界区，这些资源称为临界资源；进程互斥不允许两个以上共享临界资源的并发进程同时进入临界区。利用 **P - V** 原语和信号量可以方便地解决并发进程对临界区的进程互斥问题。

设信号量 **mutex** 是用于互斥的信号量，初值为 1，表示没有并发进程使用该临界区。于是各并发进程的临界区可改写成下列形式的代码段：

```
P(mutex);
临界区
V(mutex);
```

### (4) 用 **P - V** 操作实现进程同步

要用 **P - V** 操作实现进程同步，需要引进私用信号量，私用信号量只与制约进程和被制约进程有关，而不是与整组并发进程相关。与此相对，进程互斥使用的信号量为公用信号量。首先为各并发进程设置私用信号量，然后为私用信号量赋初值，最后利用 **P - V** 原语和私用信号量规定各进程的执行顺序。

经典同步问题的例子是生产者 - 消费者问题。这要求存后再取，取后再存，即有两个制约关系，为此，需要两个信号量，记为 **Bufempty** 和 **Bufull**，它们的初值分别是 1 和 0，相应的程序段形式是：

```
1 生产者
loop
```

```

生产一产品 next;
P(Bufempty);
next 产品存缓冲区;
V(Buffull);
endloop
I 消费者
loop
P(Buffull);
从缓冲区中取产品;
V(Bufempty);
使用产品
endloop
    
```

#### 4. 进程通信与管程

##### (1) 进程通信

通信（Communication）就是在进程间传送数据。一般来说，进程间的通信根据通信内容可以划分为两种：控制信息的传送和大批量数据的传送。把控制信息的传送称为低级通信，而把大批量数据的传送称为高级通信。进程的同步和互斥是通过信号量进行通信来实现的，属于低级通信。高级通信原语则提供两种通信方式：有缓冲区的通信和无缓冲区的通信。

##### (2) 管程

汉森（Brinch Hansen）和霍尔（Hoare）提出了一个新的同步机制——管程。管程是一个由过程、变量及数据结构等组成的集合，即把系统中的资源用数据抽象地表示出来。这样，对资源的管理就可以用数据及在其上实施操作的若干过程来表示，而代表共享资源的数据及在其上操作的一组过程就构成了管程。进程可以在任何需要资源的时候调用管程，且在任一时刻最多只有一个进程能够真正地进入管程，而其他调用进程则只能等待。由此看来，管程实现了进程之间的互斥，使临界区互斥实现了自动化，它比信号量更容易保证并发进程的正确性。管程结构如图 3-4 所示。

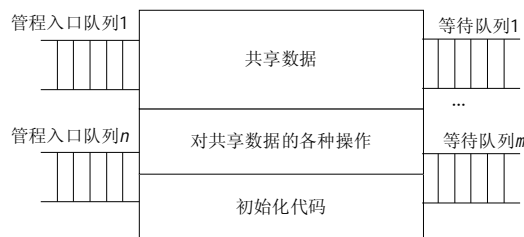


图 3-4 管程结构

#### 5. 进程调度与死锁

##### (1) 进程调度

进程调度即处理器调度（又称上下文转换），它由调度原语实现。进程调度的方式有两类：剥夺方式与非剥夺方式。所谓非剥夺方式是指一旦某个作业或进程占有了处理器，别的进程就不能把处理器从这个进程手中夺走，直到该进程自己因调用原语操作而进入阻塞状态，或时间片用完而让出处理机。剥夺方式即就绪队列中一旦有进程优先级高于当前执

行进程优先级时，便立即发生进程调度，转让处理机。

### (2) 进程调度算法

进程调度的算法是服务于系统目标的策略，对于不同的系统与系统目标，常采用不同的调度算法，如：

- I 先来先服务 (FCFS) 调度算法，又称先进先出 (FIFO)。就绪队列按先来后到原则排队。
- I 优先数调度。优先数反映了进程优先级，就绪队列按优先数排队，有两种确定优先级的方法，即静态优先级和动态优先级。静态优先级是指进程的优先级在进程开始执行前确定，执行过程中不变，而动态优先级则可以在进程执行过程中改变。
- I 轮转法 (Round Robin)。就绪队列按 FCFS 方式排队。每个进程执行一次占有处理器时间都不超过规定的时间单位 (时间片)。若超过，则自行释放自己所占有的 CPU 而排到就绪队列的末尾，等待下一次调度。同时，进程调度程序又去调度当前就绪队列中的第一个进程。

### (3) 死锁

当若干个进程互相竞争对方已占有的资源，无限期地等待，不能向前推进时，会造成“死锁”。死锁是系统的一种出错状态，应该尽量预防和避免。产生死锁的主要原因是供共享的系统资源不足、资源分配策略和进程的推进顺序不当。

产生死锁的必要条件是：互斥条件、保持和等待条件、不剥夺条件、环路等待条件。

解决死锁有两种策略：一种是在死锁发生前采用的预防和避免策略；另一种是在死锁发生后采用的检测与恢复策略。

死锁的预防主要是通过打破死锁产生的四个必要条件之一来保证不会产生死锁。死锁避免策略，则是在系统进行资源分配时，先执行一个死锁避免算法 (典型的如银行家算法)，来保证本次分配不会导致死锁的发生。实际上系统出现死锁的概率很小，故从系统所花的代价上看，采用死锁发生后的检测与恢复策略要比采用死锁发生前的预防与避免策略代价小一些。

## 6. 线程

在支持线程的操作系统中，线程是进程中的一个实体，是系统实施调度的独立单位。线程只拥有一些在运行中必不可少的资源，它与属于同一个进程的其他线程共享该进程所拥有的资源。各线程之间可以并发地运行。线程切换时只需保存和设置少量寄存器的内容，而并不涉及存储器管理方面的操作，所以线程切换的开销远远小于进程的切换 (原运行进程状态的切换还要引起资源转移及现场保护等问题)。同一个进程中的多个线程共享同一个地址空间，这使得线程之间同步和通信的实现也比较容易。

## 3.3 存储管理

存储管理的主要对象是内存，是除处理器外操作系统管理的最重要的资源，也是历年考核的重点，从 1991 年到 2003 年共有 6 道试题涉及存储管理的知识点。存储管理部分偏重于虚拟存储、分区存储、地址转换及交换技术等知识。

### 1. 存储管理的概念

存储管理主要是指对内存的管理，负责对内存的分配和回收、内存的保护和内存的扩充。存储管理的目的是尽量提高内存的使用效率。

### 2. 单一连续区管理

在单道程序系统中，内存区域的用户空间全部为一个作业或进程占用。单一连续分配方法主要用于早期单道批处理系统。单一连续分配方法主要采用静态分配方法，为降低成本和减少复杂度，通常不对内存进行保护，因而会引起冲突使系统瘫痪。

### 3. 分区存储管理

分区存储管理包括固定分区、可变分区，其基本思想是把内存划分成若干个连续区域，每个分区装入一个作业运行。要求作业一次性装入内存，且分区内部地址必须连续。

#### （1）固定分区存储管理

固定分区分配方法是把内存空间固定地划分为若干个大小不等的区域，划分的原则由系统决定。系统使用分区表描述分区情况。分区一旦划分结束，在整个执行过程中每个分区的长度和内存的总分区个数保持不变。

#### （2）可变分区存储管理

可变分区分配方法是把内存空间按用户要求动态地划分成若干个分区。随着进程的执行，剩余的自由区域会变得更小，这时需要合并自由区和存储拼接技术。合并自由区是将相邻自由存储区合并为单一自由区的方法；存储拼接技术也称碎片收集，包括移动存储器的所有被占用区域到主存的某一端。可变分区克服了固定分区分配方法中的小作业占据大分区后产生碎片的浪费问题。

#### （3）存储分配算法

常使用的四种存储分配算法介绍如下。

- ❶ 首次适应算法：把内存中的可用分区单独组成可用分区表或可用分区自由链，按起始地址递增的次序排列。每次按递增次序向后找，一旦找到大于或等于所要求内存长度的分区，则结束探索，从找到的分区中找出所要求的内存长度分配给用户，并把剩余的部分进行合并。
- ❷ 循环适应算法：上述首次适应法经常利用的是低地址空间，后面经常是较大的空白区，为使内存所有线性地址空间尽可能轮流使用到，每重新分配一次，都在当前之后寻找。
- ❸ 最佳适应算法：最佳适应算法是将输入作业放入主存中与它所需大小最接近的空白区中，使剩下的未用空间最小，该法要求空白区大小按从小到大次序组成空白区可用表或自由链。在进行分配时总是从最小的一个开始查询，因而找到了一个能满足要求的空白区便是最佳的一个。
- ❹ 最差适应算法：分配时把一个作业程序放入主存中最不适合它的空白区，即最大的空白区（空闲区）内。

#### （4）交换与覆盖技术

覆盖与交换技术是在多道程序环境下用来扩充内存的两种方法。覆盖技术主要用在早期的操作系统中，而交换技术则在现代操作系统中得到了进一步发展。

覆盖技术是一种解决小内存运行大作业的方法。一个作业中若干程序段和数据段可以

不同时使用，这样它们就可以共享内存的某个区域，再根据需要分别调入该区域，这个区域就称为覆盖区。将程序执行时并不要求同时装入主存的覆盖组成一组，并称其为覆盖段，这个覆盖段分配到同一个覆盖区。

交换技术可以将暂不需要的作业移到外存，让出内存空间以调入其他作业，交换到外存的作业也可以被再次调入。交换技术与覆盖技术相比不要求给出程序段之间的覆盖结构。交换主要是在作业之间进行的，而覆盖则主要是在同一个作业内进行的。

#### 4. 页式存储管理

分页的基本思想是把程序的逻辑空间和内存的物理空间按照同样的大小划分成若干页面，以页面为单位进行分配。在页式存储管理中，系统中虚地址是一个有序对（页号，位移）。系统为每一个进程建立一个页表，其内容包括进程的逻辑页号与物理页号的对应关系、状态等。

#### 5. 段式存储管理

段式存储管理与页式存储管理相似。分段的基本思想是把用户作业按逻辑意义上有完整意义的段来划分，以段为单位作为内、外存交换的空间尺度。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。在分段系统中，容许程序（作业）占据内存中许多分离的分区。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寄存器，判定访问地址是否越界也困难了。在分段存储系统中常常利用存储保护键实现存储保护。分段系统中虚地址是一个有序对（段号，位移）。系统为每个作业建立一个段表，其内容包括段号、段长、内存起始地址和状态等。状态指出这个段是否已调入内存，即内存起始地址指出这个段，状态指出这个段的访问权限。

#### 6. 段页式存储管理

段页式管理是段式和页式两种管理方法结合的产物，综合了段式组织与页式组织的特点，根据程序模块分段，段内再分页，内存被划分成定长的页。段页式系统中虚地址形式是（段号、页号、位移）。系统为每个进程建立一个段，为每个段建立一个页表。段页式管理采用段式分配、页式使用的方法，便于动态连接和存储的动态分配。这种存储管理能提高内存空间的利用率。段页式虚拟存储管理结合了段式和页式的优点，但增加了设置表格（段表、页表）和查表等开销，段页式虚拟存储器一般只在大型计算机系统中使用。

#### 7. 页面调度

如果选择的页面被频繁地装入和调出，这种现象称为“抖动”，应减少和避免抖动现象。常用的页面调度算法有以下几种。

(1) 最优（OPT）算法。选择不再使用或最远的将来才被使用的页，难以实现，常用于淘汰算法的比较。

(2) 随机（RAND）算法。随机地选择被淘汰的页，开销小，但是可以选中立即就要访问的页。

(3) 先进先出（First in First out, FIFO）算法，又称轮转法（RR）。选择在内存驻留时间最长的页，似乎合理，但可能淘汰掉频繁使用的页。另外，使用 FIFO 算法时，在未给予进程分配足够的页面数时，有时会出现给予进程的页面数增多，缺页次数反而增加的异常现象。FIFO 算法简单，可采用队列实现。

(4) 最近最少使用（Least Recently Used 缩写为 LRU）算法。选择离当前时间最近的

一段时间内使用得最少的页。这个算法的主要出发点是，如果某个页被访问了，则它可能马上就要被访问；反之，如果某个页长时间未被访问，则它在最近一段时间也不会被访问。

另外，还有最不经常使用的页面先淘汰（LFU, least frequent used）、最近没有使用的页面先淘汰（NUR）、最优淘汰算法（OPT, optimal replacement algorithm）等。

### 8. 存储管理方式比较

存储管理方式的比较如表 3-1 所示。

表 3-1 存储管理方式比较表

方法 功能	单一连续区	分区式		页式		段式	段页式
		固定分区	可变分区	静态	动态		
适用环境	单道	多道		多道		多道	多道
虚拟空间	一维	一维		一维		二维	二维
重定位	静态	静态	动态	动态		动态	动态
分配方式	静态分配连续区	静态或动态分配连续区		静态或动态以页为单位分配非连续区		动态以段为单位分配非连续区	动态以页为单位分配非连续区
释放	执行完成后全部释放	执行完成后全部释放	分区释放	执行完成后释放	淘汰与执行完成后释放	淘汰与执行完成后释放	淘汰与执行完成后释放
保护	越界保护或没有	越界保护与保护键		越界保护与控制前保护		同左	同左
内存扩充	覆盖与交换技术	同左		同左	外存、内存统一管理的虚存	同左	同左
共享	不能	不能		较难		方便	方便
硬件支持	保护用寄存器	保护用寄存器和重定位机构		地址变换寄存、中断机构、保护机构		段式地址变换机构，保护与中断，动态连接结构	同左

## 3.4 设备管理

设备管理是指操作系统对除 CPU 和内行之外所有设备的管理。历年考题中主要涉及缓冲技术和 Spooling 技术等知识点。

### 1. 设备管理的概念

在计算机系统中，除了处理器和内存之外，其他的大部分硬设备称为外部设备。它包括输入/输出设备、辅存设备及终端设备等。为了完成上述主要任务，设备管理程序一般要提供下述功能。

(1) 提供和进程管理系统的接口。当进程要求设备资源时，该接口将进程要求转达给设备管理程序。

(2) 进行设备分配。按照设备类型和相应的分配算法把设备和其他有关的硬件分配给

请求该设备的进程，并把未分配到所请求设备或其他有关硬件的进程放入等待队列。

(3) 实现设备和设备、设备和 CPU 等之间的并行操作。

(4) 进行缓冲区管理。主要减少外部设备和内存与 CPU 之间的数据速度不匹配的问题，系统中一般设有缓冲区（器）来暂放数据。设备管理程序负责进行缓冲区分配、释放及有关的管理工作。

## 2. 数据传输控制方式

外围设备和内存之间的常用数据传送控制方式介绍如下。

(1) 程序控制方式。

(2) 中断方式。

(3) 直接存储访问（DMA）方式。指外部设备和内存之间开辟直接的数据交换通路。

(4) 通道方式。通道又称为输入/输出处理器（IOP），主要有三类通道：字节多路通道、选择通道和成组多路通道。

## 3. 设备的分配

(1) 设备分配原则

设备分配方式有两种：一种是静态分配；另一种是动态分配。静态分配方式是在用户作业开始执行之前，由系统一次分配该作业所要求的全部设备、控制器和通道。一旦分配之后，这些设备、控制器和通道就一直为该作业所占用，直到该作业被撤销。静态分配方式不会出现死锁，但是设备的使用效率低。

动态分配在进程执行过程中根据执行需要进行。当进程需要设备时，通过系统调用命令向系统提出设备请求，由系统按照事先规定的策略给进程分配所需要的设备、I/O 控制器和通道，一旦用完之后，便立即释放。动态分配方式有利于提高设备的利用率，但如果分配算法使用不当，则有可能造成进程死锁。

(2) 设备分配策略

常用的分配策略有先请求先分配、优先级高者先分配策略等。

(1) 先请求先分配。

(2) 优先级高者先分配。这种策略和进程调度的优先级算法是一致的，即进程的优先级高，那么它的 I/O 请求也优先满足。对于相同优先级的进程来说，则按照先请求先分配策略分配。

## 4. 磁盘调度算法

访问磁盘的时间因子由三部分构成，它们是查找（查找磁道）时间、等待（旋转等待扇区）时间和数据传输时间，其中查找时间是决定因素。因此，磁盘调度算法主要有以下几种。

(1) 先来先服务（FCFS）调度。按先来后到次序服务，未做优化。

(2) 最短查找时间优先（SSTF）调度。SSTF 查找距离磁头最短（也就是查找时间最短）的请求作为下一次服务的对象。SSTF 查找模式有高度局部化的倾向，会推迟一些请求的服务，甚至引起无限拖延（又称饥饿）。

(3) SCAN 调度。又称电梯算法，SCAN 算法是磁头前进方向 L 的最短查找时间优先算法，它排除了磁头在盘面局部位置上的往复移动，SCAN 算法在很大程度上消除了 SSTF 算法的不公平性，但仍有利于对中间磁道的请求。

### 5. 虚设备与 Spooling 技术

Spooling (simultaneous peripheral operation on line) 的意思是外部设备同时联机操作, 又称为假脱机输入输出操作, 采用一组程序或进程模拟一台输入输出处理器。Spooling 系统的组成如图 3-5 所示。该技术利用了专门的外围控制机将低速 I/O 设备上的数据传送到高速设备上, 或者相反。但是当引入多道程序后, 完全可以利用其中的一道程序来模拟脱机输入时的外围控制机的功能, 把低速的 I/O 设备上的数据传送到高速磁盘上; 再利用另一道程序来模拟脱机输出时的外围控制机的功能, 把高速磁盘上的数据传送到低速的 I/O 设备上。这样便可以在主机的控制下实现脱机输入、输出的功能。此时的外围操作与 CPU 对数据的处理同时进行, 我们将这种在联机情况下实现的同时外围操作称为 Spooling, 或称为假脱机操作。

采用假脱机技术, 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 系统必须有高速、大容量并且可随机存取的外存 (如磁盘或磁鼓) 支持。

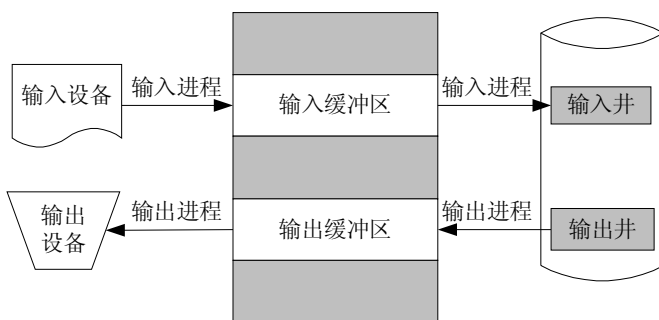


图 3-5 Spooling 系统示意图

## 3.5 文件管理

与存储管理相对, 文件管理是对外部存储设备上的以文件方式存放的信息的管理。核心内容是文件的结构和访问方式、存储空间管理及目录结构等知识点。从历年试题看, 有涉及存储空间管理和与 UNIX 具体的文件系统相关的试题。

### 1. 文件和文件系统的概念

#### (1) 文件和文件系统

文件是信息的一种组织形式, 是存储在辅助存储器上的具有标识名的一组信息集合。它可以是有结构的, 也可以是无结构的。操作系统中由文件系统来管理文件的存储、检索、更新、共享和保护。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。

#### (2) 文件类型

根据不同的方面, 文件有多种分类方法:

- I 按文件的用途可以分为系统文件、库文件和用户文件等。

- | 按文件的安全属性可分为只读文件、读写文件、可执行文件和不保护文件等。
- | 按文件的信息流向可以分为输入文件、输出文件和输入/输出文件等。
- | 按文件的组织形式可以分为普通文件、目录文件和特殊文件等。特殊文件是 UNIX 系统采用的技术，把所有的输入输出设备都视为文件（特殊文件）。特殊文件的使用形式与普通文件相同。

## 2. 文件的结构和存取方式

### (1) 文件的结构

文件的结构是指文件的组织形式，从用户观点所看到的文件组织形式，称为文件的逻辑结构。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。记录文件由记录组成，即文件内的信息划分成多个记录，以记录为单位组织和利用信息。记录文件有顺序文件、索引顺序文件、索引文件和直接文件。

文件的物理结构是指文件在存储设备上的存放方法。文件的物理结构侧重于提高存储器的利用效率和降低存取时间。文件的存储设备通常划分为大小相同的物理块，物理块是分配和传输信息的基本单位。文件的物理结构涉及文件存储设备的组织策略和文件分配策略，决定文件信息在存储设备上的存储位置。常用的文件分配策略有顺序分配（连续分配）、链接分配（串联分配）、索引分配。

### (2) 文件的访问方式

用户通过对文件的访问（读写）来完成对文件的查找、修改、删除和添加等操作。常用的访问方法有两种，即顺序访问和随机访问。

## 3. 文件存储设备管理

文件存储设备管理，就是操作系统要有效地进行存储空间的管理。由于文件存储设备是分成许多大小相同的物理块，并以块为单位交换信息，因此，文件存储设备的管理实质上是对空闲块的组织和管理问题，它包括空闲块的组织、空闲块的分配与空闲块的回收等问题。有三种不同的空闲块管理方法，它们是索引法、链接法和位图法。

## 4. 文件控制块和文件目录

### (1) 文件控制块

文件控制块是系统和管理文件时所必需的信息的数据结构，是文件存在的惟一标志，简称为 FCB。文件目录就是文件控制块的有序集合。FCB 的内容包括相应文件的基本属性，大致可以分成四个部分。

- | 基本信息：如文件名、文件类型和文件组织等；
- | 保护信息：如口令、所有者名、保存期限和访问权限等；
- | 位置信息：如存储位置、文件长度等；
- | 使用信息：如时间信息、最迟使用者等。

### (2) 文件目录

文件控制块的集合称为文件目录，文件目录也被组织成文件，常称为目录文件。

文件管理的一个重要方面是对文件目录进行组织和管理。文件系统一般采用一级目录结构、二级目录结构和多级目录结构。DOS、UNIX、Windows 系统都是采用多级（树型）目录结构。

## 5. 文件的操作与使用

### （1）文件的使用

一般文件系统提供一组专门用于文件、目录的管理命令。如目录管理、文件控制和文件存取等命令。

- Ⅰ 目录管理命令：如建立目录、显示工作目录、改变目录、删除目录。
- Ⅰ 文件控制命令：如建立文件、删除文件、打开文件、关闭文件、改文件名、改变文件属性。
- Ⅰ 文件存取命令：如读写文件、显示文件内容、复制文件等。

### （2）文件共享和安全

文件的共享是指不同的用户使用同一文件。文件的安全是指文件的保密和保护，即限制非法用户使用和破坏文件。

文件的共享可以采用文件的绝对路径名或相对路径名共享同一文件。一般的文件系统要求用户先打开文件，再对文件进行读写，不再使用时关闭文件。若两个用户可以同时打开文件，对文件进行存取，称为动态文件共享。

文件的安全管理措施常常在系统级、用户级、目录级和文件级上实施。

## 3.6 作业管理

作业管理包含了进程管理，作业管理的理解可以加强对进程管理的理解。历年试题中常将作业调度与进程调度、设备管理综合起来考查，所以注意对操作系统整体的工作机制的理解和掌握。

### 1. 作业状态与作业管理

操作系统中用来控制作业的进入、执行和撤销的一组程序称为作业管理程序，这些控制功能也能通过把作业细化，通过进程的执行来实现。

#### （1）作业状态及其转换

一个作业从交给计算机系统到执行结束退出系统，一般都要经历提交、后备、执行和完成四个状态。其状态转换如图 3-6 所示。

- Ⅰ 提交状态。作业由输入设备进入外存储器（也称输入井）的过程称为提交状态。处于提交状态的作业，其信息正在进入系统。
- Ⅰ 后备状态。当作业的全部信息进入外存后，系统就为该作业建立一个作业控制块（JCB）。
- Ⅰ 执行状态。一个后备作业被作业调度程序选中分配了必要的资源并进入了内存，作业调度程序同时为其建立了相应的进程后，该作业就由后备状态变成了执行状态。



道，即操作系统的接口；另一种是指这种交互环境的控制方式，即操作环境。

#### （1）操作系统的接口

操作系统的接口又可分成命令接口和程序接口。

Ⅰ 命令接口：命令接口包含键盘命令和作业控制命令；

Ⅰ 程序接口：程序接口又称为编程接口或系统调用，程序经编程接口请求系统服务，即通过系统调用，程序与操作系统通信。

系统调用是操作系统提供给编程人员的惟一接口。系统调用对用户屏蔽了操作系统的具体动作而只提供有关功能。系统调用大致分为设备管理、文件管理、进程控制、进程通信和存储管理等。

#### （2）操作环境

操作环境支持命令接口和程序接口，提供友好的、易用的操作平台。操作系统的交互界面已经从早期的命令驱动方式，发展到菜单驱动方式、图符驱动方式和视窗操作环境。

## 3.7 网络操作系统

历年试题中常将操作系统实例与操作系统原理结合起来考查，包括 UNIX 中的文件系统存储管理和 Shell、Windows NT 中的线程等概念。

### 1. UNIX 系统

#### （1）系统结构

UNIX 是一个分时操作系统。它利用最内层硬件提供的基本服务，向外层提供全部应用程序所需要的服务。UNIX 系统结构可分为三层，即最内层的 UNIX 操作系统核心，它包含文件管理子系统和进程控制子系统两个部分，进程控制子系统负责进程的创建、撤销、同步、通信、进程调度及存储管理；文件管理子系统负责对系统中的文件进行管理并提供高速缓冲机制。最外层是用户程序，包括许多应用软件。中间层则是 Shell 命令解释层、实用程序、库函数等，该层中的 Shell 解释程序是用户和 UNIX 的操作界面。

#### （2）文件系统

UNIX 中的文件可以分为三类，普通文件、目录文件和设备文件。

进程通过系统调用要求文件系统提供服务。例如 Open，打开一个文件，准备读、写；Close，关闭文件；Write，写文件；Read，读文件；Stat，查询一个文件的属性；Chown，改变文件所有者；Chmod，改变文件存取许可权；Create，创建一个普通文件；Mkdir，创建一个目录文件；Cd，改变当前目录；Link，建立连接；Unlink，删除文件连接。

#### （3）进程管理和存储管理

UNIX 系统中进程控制子系统负责进程的创建、撤销、同步、通信、进程调度及存储管理。

Ⅰ 系统控制：用于控制进程的系统调用有：Fork，创建一个子进程；Exec，改变执行程序的映像；Exit，结束一个进程的执行；Wait，暂停进程的执行，用于进程之间的同步，例如父进程等待子进程执行结束；Signal，控制进程对特别事件的响应；Kill，发送软中断信号；Msgsnd，发送消息；Msgrcv，接收消息。

- I 进程调度方法：UNIX 系统中的进程调度采用动态优先数的算法。
- I 存储管理：UNIX 采用段页式管理。分配的单位是块，如每块 81（字节）。UNIX 采用了 LRU（最近最少使用算法）作为虚存中的淘汰算法。同时 UNIX 还使用交换技术和对换与请求调页混合的技术，可以有效地避免颠簸现象的出现。

#### (4) Shell

UNIX 系统中的命令解释程序称做 Shell。命令行的一般形式是：命令名[参数, 参数, …]。命令行可以带若干个参数，也可以不带参数。一个命令名对应于一个可执行文件。Shell 在收到命令以后，查找相应文件，将其调入内存并且执行，命令行中的参数在相应程序执行时加以使用。

Shell 向用户提供了重定向操作与管道操作。UNIX 系统中有三个标准文件：标准输入文件、标准输出文件和标准错误文件，分别为 0 号文件、1 号文件和 2 号文件。而且这些文件可以独立地被重新定向。例如命令 `ls` 把工作目录中的内容显示在标准输出文件上（即屏幕）。但命令：`ls>Outfile`，则将 `ls` 的输出重定向到当前目录下的文件 `Outfile` 上（若不存在，则创建此文件）。“>”是标准输出重定向符号。标准输入重定向的符号为“<”，标准错误重定向的符号为“2>”。另外，“>>”表示标准输出重定向且拼接。

管道是 UNIX 系统的一个重要特色，它允许在生产者进程与消费者进程之间传递数据流。管道机制建立了一个临时的“哑”文件（管道文件），把生产者进程的标准输出重定向到管道文件上，并启动消费者进程，将它的标准输入重定向到管道文件上（即从管道文件读），从而把两个进程的输出、输入连接起来。管道的记号为“|”。使用管道机制可以简化命令行的写法，例如三个连续的命令行：

```
ls > file1
sort < file1 > file2
pr < file2
```

可以表示为：

```
ls | sort | pr
```

此外，UNIX 系统还提供后台命令，例如：

```
cc prog.c &
```

表示用户不必等到对 `prog.c` 编译结束就可键入新的命令，Shell 可以响应并处理新的命令，从而向用户提供了多个任务的并行。

## 2. Windows NT 系统

Windows NT 基于客户/服务器模式，采用微内核技术，支持对称多处理结构，支持多线程并行，支持多种硬件平台和多种网络协议。它采用 16 位标准字符集的统一代码方式（Unicode 代码），支持国际字符集，提供性能良好的文件系统。

### (1) 系统结构

Windows NT 是一种对象式操作系统，所有的系统资源都被表示为对象。Windows NT 采用层次式与客户/服务器模式相结合的系统结构。Windows NT 共分三层，即用户模式、核心和硬件抽象层。

### (2) 进程与线程

在 Windows NT 执行体中，进程是由对象管理器创建和删除的对象。每个进程创建时

只有一个线程。根据需要，进程可以创建更多的线程。大多数进程是用户态进程，用户态进程通过系统调用访问操作系统。当核心态完成服务，在将控制权交还用户程序前将线程切换回用户态。应用程序在用户态下运行，Windows NT 保护子系统也是如此。后者是提供重要的操作系统功能的用户态服务进程。

多线程进程实现的并行避免了进程之间并行的缺点：创建线程的开销比创建进程的开销要小，同一进程的线程共享进程的地址空间，所以线程的切换（处理器调度）比进程的切换快。

Windows NT 内核采用优先级的方案选定线程执行的次序。高优先级线程先于低优先级线程执行。内核周期性地改变线程的优先级，以确保所有线程均能执行。

线程是进程的活动成分，它可以共享进程的资源与地址空间，通过线程的活动，进程可以同时提供多种服务（对服务器进程而言）或实行子任务并行（对用户进程而言）。每个 Windows NT 进程创建时只有一个线程，根据需要在运行过程中创建更多的线程（前者亦可称“主线程”）。显然，只有主线程的进程就是传统意义下的进程。内核负责线程的调度。线程的优先级可以动态地改变。采用线程机制的最大优点是节省开销。

## 3.8 嵌入式操作系统

### 3.8.1 嵌入式操作系统概述

嵌入式系统通常是指内部包含智能控制器的设备，它具有集成度高、体积小、反应速度快、智能化、稳定及可靠性强等特点。嵌入式实时控制系统必须要非常仔细地研究实时性的保证实施。

嵌入式系统应具有的特点是高可靠性；在恶劣的环境或突然断电的情况下，系统仍然能够正常工作；许多嵌入式应用要求实时性，这就要求嵌入式操作系统具有实时处理能力；嵌入式系统和具体应用有机地结合在一起，它的升级换代也是和具体产品同步进行的；嵌入式系统中的软件代码要求高质量、高可靠性，一般都固化在只读存储器或闪存中，也就是说软件要求固态化存储，而不是存储在磁盘等载体中。

嵌入式软件（Embedded Software），从广义上讲是计算机软件的一种，它也是由程序及其文档组成的。嵌入式软件是嵌入在设备内部并控制设备行为的一种专用软件，其最基本的特点是软件固态化存储在存储芯片或单片机中，而不是存储于磁盘等载体中。嵌入式软件一般在设备启动时自动运行，无需人工干预。通常要求具有实时响应能力，一般不要求复杂的用户界面，也不需用户进行二次开发。

嵌入式软件可分成系统软件、支撑软件（中间件）、应用软件三类，最低层即系统软件，包括操作系统及数据库管理系统。下面定义的嵌入式操作系统、嵌入式数据库、嵌入式中间件、嵌入式应用软件，必须同时符合上述嵌入式软件的定义。

嵌入式操作系统 EOS（Embedded Operating System）是以应用为中心，以计算机技术为基础，软硬件可裁减，对功能、可靠性、成本、体积、功耗有严格要求的专用性计算机系统。嵌入式操作系统是设备信息系统的核心，管理、监控和维护设备硬件和软件资源，支持和调度各种应用软件的运行，实现处理机管理、内存管理、I/O 设备管理、文件管理及

作业管理。

目前，已推出一些应用比较成功的 EOS 产品系列。随着 Internet 技术的发展、信息家电的普及应用及 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。嵌入式操作系统在系统实时高效性、硬件的相关依赖性、软件固化及应用的专用性等方面具有较为突出的特点。EOS 是相对于一般操作系统而言的，它除了具备一般操作系统最基本的功能，如任务调度、同步机制、中断处理、文件处理等，还有以下特点。

(1) 可装卸性。开放性、可伸缩性的体系结构。

(2) 强实时性。EOS 实时性一般较强，可用于各种设备控制当中。

(3) 统一的接口。提供各种设备驱动接口。

(4) 操作方便、简单，提供友好的图形界面，追求易学易用。

(5) 提供强大的网络功能，支持 TCP/IP 协议及其他协议，提供 TCP/UDP/IP/PPP 协议支持及统一的 MAC 访问层接口，为各种移动计算设备预留接口。

(6) 强稳定性，弱交互性。嵌入式系统一旦开始运行就不需要用户过多地干预，这就要求负责系统管理的 EOS 具有较强的稳定性。嵌入式操作系统的用户接口一般不提供操作命令，它通过系统的调用命令向用户程序提供服务。

(7) 固化代码。在嵌入式系统中，嵌入式操作系统和应用软件被固化在嵌入式系统计算机的 ROM 中。辅助存储器在嵌入式系统中很少使用，因此，嵌入式操作系统的文件管理功能应该能够很容易地拆卸，而用各种内存文件系统。

(8) 更好的硬件适应性，也就是良好的移植性。

国际上用于信息电器的嵌入式操作系统有 40 种左右。现在，市场上非常流行的 EOS 产品，包括 3Com 公司下属子公司的 Palm OS，Microsoft 公司的 Windows CE 和开放源代码的 Linux。

### 3.8.2 常用的嵌入式操作系统

#### 1. Palm OS

Palm OS 是一种 32 位的嵌入式操作系统。Palm 提供了串行通信接口和红外线传输接口，利用它可以方便地与其他外部设备通信、传输数据；拥有开放的 OS 应用程序接口，开发商可根据需要自行开发所需的应用程序。Palm OS 是一套具有强开放性的系统，现在有大数千种专门用 Palm OS 编写的应用程序，从程序内容上看，小到个人管理、游戏，大到行业解决方案。

Palm OS 是一套专门为掌上电脑开发的 OS。在编写程序时，Palm OS 充分考虑了掌上电脑内存相对较小的情况，因此它只占有非常小的内存。由于基于 Palm OS 编写的应用程序占用的空间也非常小，所以，基于 Palm OS 的掌上电脑（虽然只有几兆字节的 RAM）可以运行众多应用程序。

由于 Palm 产品的最大特点是使用简便、机体轻巧，因此决定了 Palm OS 应具有以下特点。

(1) 操作系统的节能功能。由于掌上电脑要求使用的电源尽可能小，因此在 Palm OS 的应用程序中，如果没有事件运行，则系统设备进入半休眠的状态；如果应用程序停止活动一段时间，则系统自动进入休眠状态。

(2) 合理的内存管理。Palm 的存储器全部是可读写的快速 RAM，动态 RAM 类似于 PC 机上的 RAM，它为全局变量和其他不需永久保存的数据提供临时的存储空间；存储 RAM 类似于 PC 机上的硬盘，可以永久保存应用程序和数据。

(3) Palm OS 的数据是以数据库的格式来存储的，为保证程序处理速度和存储器空间，在处理数据的时候，Palm OS 不是把数据从存储堆拷贝到动态堆后再进行处理，而是在存储堆中直接处理。

(4) Palm OS 与同步软件结合可以使掌上电脑与 PC 机上的信息实现同步，把台式机的功能扩展到了掌上电脑上。

## 2. Windows CE

Windows CE 也是一个开放的、可升级的 32 位嵌入式操作系统，是基于掌上电脑类的电子设备操作。Windows CE 的图形用户界面相当出色。其中，CE 中的 C 代表袖珍 (Compact)、消费 (Consumer)、通信能力 (Connectivity) 和伴侣 (Companion)；E 代表电子产品 (Electronics)。与 Windows 95/98、Windows NT 不同的是，Windows CE 是所有源代码全部由微软自行开发的嵌入式新型操作系统，其操作界面虽来源于 Windows 95/98，但 Windows CE 是基于 Win32 API 重新开发的、新型的信息设备平台。Windows CE 具有模块化、结构化和基于 Win32 应用程序接口及与处理器无关等特点。Windows CE 不仅继承了传统的 Windows 图形界面，并且在 Windows CE 平台上可以使用 Windows 95/98 上的编程工具（如 Visual Basic、Visual C++ 等），使用同样的函数，使用同样的界面网格，使绝大多数的应用软件只需简单修改和移植就可以在 Windows CE 平台上继续使用。

Windows CE 的设计目标是：模块化及可伸缩性、实时性能好，通信能力强大，支持多种 CPU。它的设计可以满足多种设备的需要，这些设备包括了工业控制器、通信集线器及销售终端之类的企业设备，还有像照相机、电话和家用娱乐器材之类的消费产品。一个典型的基于 Windows CE 的嵌入系统通常为某个特定用途而设计，并在不联机的情况下工作。它要求所使用的操作系统体积较小，内有对中断的响应功能。

Windows CE 有如下特点。

(1) 具有灵活的电源管理功能，包括睡眠/唤醒模式。

(2) 使用了对象存储技术，包括文件系统、注册表及数据库。它还具有很多高性能、高效率的操作系统特性，包括按需换页、共享存储、交叉处理同步、支持大容量堆等。

(3) 拥有良好的通信能力。广泛支持各种通信硬件，亦支持直接的局域连接及拨号连接，并提供与 PC、内部网及因特网的连接，还提供与 Windows 9x/NT 的最佳集成和通信。

(4) 支持嵌套中断。允许更高优先级别的中断首先得到响应，而不是等待低级别的 ISR 完成。这使得该操作系统具有嵌入式操作系统所要求的实时性。

(5) 更好的线程响应能力。对高级别 IST（中断服务线程）的响应时间上限的要求更加严格，在线程响应能力方面的改进，帮助开发人员掌握线程转换的具体时间，并通过增强监控能力和对硬件的控制能力帮助他们创建新的嵌入式应用程序。

(6) 256 个优先级别。可以使开发人员在控制嵌入式系统的时序安排方面有更大的灵活性。

(7) Windows CE 的 API 是 Win32 API 的一个子集，支持近 1 500 个 Win32 API。有了这些 API，足可以编写任何复杂的应用程序。当然，在 Windows CE 系统中，所提供的

API 也可以随具体应用的需求而定。

### 3. Linux

Linux 是一个类似于 UNIX 的操作系统，Linux 系统不仅能够运行于 PC 平台，还在嵌入式系统方面大放光芒，在各种嵌入式 Linux OS 迅速发展的状况下，Linux OS 逐渐形成了可与 Windows CE 等 EOS 进行抗衡的局面。嵌入式 Linux OS 的特点如下。

- (1) 精简的内核，性能高、稳定，多任务。
- (2) 适用于不同的 CPU，支持多种体系结构，如 X86、ARM、MIPS、ALPHA、SPARC 等。
- (3) 能够提供完善的嵌入式 GUI 及嵌入式 X-Windows。
- (4) 提供嵌入式浏览器、邮件程序、MP3 播放器、MPEG 播放器、记事本等应用程序。
- (5) 提供完整的开发工具和 SDK，同时提供 PC 上的开发版本。
- (6) 用户可定制，可提供图形化的定制和配置工具。
- (7) 常用嵌入式芯片的驱动集，支持大量的周边硬件设备，驱动丰富。
- (8) 针对嵌入式的存储方案，提供实时版本和完善的嵌入式解决方案。
- (9) 完善的中文支持，强大的技术支持，完整的文档。
- (10) 开放源码，丰富的软件资源，广泛的软件开发者的支持，价格低廉，结构灵活，适用面广。

### 4. 三种嵌入式操作系统的比较

嵌入式 Linux OS 与 Windows CE 相比的优点如下。

(1) Linux 是开放源代码的，不存在黑箱技术，遍布全球的众多 Linux 爱好者都是 Linux 开发者的强大技术支持者；而 Windows CE 是非开放性 OS，使第三方很难实现产品定制。

(2) Linux 的源代码随处可得，注释丰富，文档齐全，易于解决各种问题。

(3) Linux 的内核小、效率高；而 Windows CE 在这方面是笨拙的，占用过多的 RAM，应用程序庞大。

(4) Linux 是开放源代码的 OS，在价格上极具竞争力，适合中国国情。Windows CE 的版权费用是厂家不得不考虑的因素。

(5) Linux 不仅支持 x86 芯片，还是一个跨平台的系统。到目前为止，它可以支持 20~30 种 CPU，很多 CPU（包括家电业的芯片）厂商都开始做 Linux 的平台移植工作，而且移植的速度远远超过 Java 的开发环境。如果今天采用 Linux 环境开发产品，那么将来更换 CPU 时就不会遇到更换平台的困扰。

(6) Linux 内核的结构在网络方面是非常完整的，它提供了对包括十兆位、百兆位及千兆位的以太网，还有无线网络、Token ring（令牌环）和光纤甚至卫星的支持。

(7) Linux 在内核结构的设计中考虑适应系统的可裁减性的要求，Windows CE 在内核结构的设计中并未考虑适应系统的高度可裁减性的要求。

嵌入式 Linux OS 与 Windows CE 相比的弱点如下。

(1) 开发难度较高，需要很高的技术实力。

(2) 核心调试工具不全，调试不太方便，尚没有很好的用户图形界面。

(3) 与某些商业 OS 一样，嵌入式 Linux 占用较大的内存，当然，人们可以去掉部分

无用的功能来减小使用的内存，但是如果不仔细，将引起新的问题。

(4) 有些 Linux 的应用程序需要虚拟内存，而嵌入式系统中并没有或不需要虚拟内存，所以并非所有的 Linux 应用程序都可以在嵌入式系统中运行。

3Com 公司的 Palm OS 是掌上电脑市场中较为优秀的嵌入式操作系统，是针对这一市场专门设计的系统。它有开放的操作系统应用程序接口 (API)，支持开发商根据需要自行开发所需的应用程序，具有十分丰富的应用程序。在掌上电脑市场上独占其霸主地位已久。

从技术层面上讲，Palm OS 是一套专门为掌上电脑开发的操作系统，具有许多 Windows CE 无法比拟的优势；Windows CE 过于臃肿，不适合应用在廉价的掌上电脑中。

Palm OS 是一套具有极强开放性的系统。开发者向用户免费提供 Palm OS 的开发工具，允许用户利用该工具在 Palm OS 基础上方便地编写、修改相关软件。与之相比，Windows CE 的开发工具就显得复杂多了，这使得一般用户很难掌握。这也是 Palm OS 与 Windows CE 的另一个主要区别。

Palm OS、Windows CE、Linux 这三种嵌入式操作系统各有不同的特点，不同的用途；但 Linux 比 Palm OS 和 Windows CE 更小、更稳定，而且 Linux 是开放的 OS，在价格上极具竞争力。

### 3.9 例题分析

#### 例题 1 (2003 年试题 22 ~ 26)

(1) 在 UNIX 操作系统中，当用户执行如下命令：

```
link("/user/include/myfile.sh", "/usr/userwang/youfile.sh")
```

则文件 "/usr/userwang/youfile.sh" 存放在 (22)。

(2) 假设在系统中一个文件有两个名字，它与一个文件保存两个副本的区别是 (23)。

(3) 在某超市里有一个收银员，且同时最多允许有  $n$  个顾客购物，我们可以将顾客和收银员看成是两类不同的进程，且工作流程如图 3-7 所示。为了利用 PV 操作正确地协调这两类进程之间的工作，设置了三个信号量  $S_1$ 、 $S_2$  和  $S_n$ ，且初值分别为 0、0 和  $n$ 。这样图中的 a 应填写 (24)，图中的 b1、b2 应分别填写 (25)，图中的 c1、c2 应分别填写 (26)。

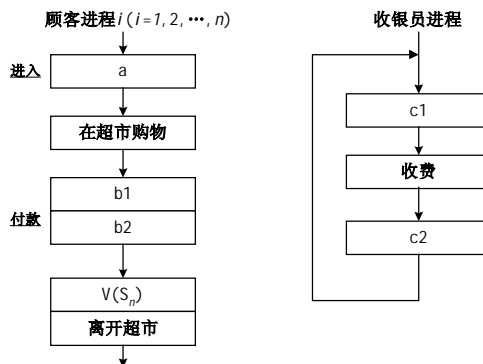


图 3-7 超市购物流程图

供选择的答案:

- (22) A. user 目录文件中                      B. include 目录文件中  
       C. userwang 目录文件中                D. youfile.sh 的文件内容中
- (23) A. 前者比后者所占用的存储空间更大  
       B. 前者需要两个目录项, 后者只需要一个目录项  
       C. 前者存取文件的速度快, 后者存取文件的速度慢  
       D. 前者改变与某个名字相联系的文件时, 另一个名字相连的文件也改变; 后者的另一个副本不改变
- (24) A. P(S1)                      B. P(S2)                      C. P(Sn)                      D. P(Sn)、 P(S1)
- (25) A. P(Sn)、 V(S2)            B. P(Sn)、 V(S1)            C. P(S2)、 V(S1)            D. V(S1)、 P(S2)
- (26) A. P(S1)、 V(S2)            B. P(Sn)、 V(S1)            C. P(S2)、 V(S1)            D. V(S1)、 P(S2)

例题分析:

第(1)小题主要考查系统操作方面的知识。

从历年的试题看, 在系统操作方面偏重于 UNIX 操作系统, 已考查了 UNIX 的输入输出重定向和管道方面的知识, 这次是 UNIX 文件系统操作方面的知识。UNIX 文件系统的常用操作命令见基础知识点 3.7 节。

本题中, Link 命令的功能是为一个文件创建链接文件。在 UNIX 系统中, 不同进程可以用系统调用 Link 原语来链接非目录文件, 从而可以直接共享该非目录文件。而且不同的用户还可以对此文件使用不同的文件名。

调用格式: Link oldname, newname

调用描述: 为一个已存在的文件 oldname 创建一个名为 newname 的链接(别名)。与此相对应的还有取消链接原语 unlink。故(22)应选 C。

第(2)小题主要考查系统操作方面的知识。

一个文件有两个名字, 实际上一个为文件, 另一个为文件的快捷方式; 一个文件保存有两个副本, 实际上是一个文件的两份拷贝。故(23)选 D。

第(3)小题主要考核进程互斥与同步机制方面的知识。解题的关键是分析哪些资源需要同步, 哪些资源需要互斥, P-V 操作的顺序非常重要。

并发进程间的基本关系是同步与互斥。进程互斥是一组并发进程中一个或多个程序段因共享某一公有资源而导致的, 它们必须以一个不允许交叉执行的单位执行。也就是说互斥是要保证临界资源在某一时刻只被一个进程访问。进程同步是指合作进程间在执行速度上要互相协调, 如生产者-消费者进程, 消费者进程必须在收到生产者进程发送的消息后才能进行消费, 否则只有等待, 直到生产者生产产品后再发送消息为止。进程互斥是指逻辑上没有什么联系的进程, 在竞争临界资源时产生的一种制约关系, 即一个进程使用资源时, 其他进程只能等待。

操作系统的重要功能是解决进程间的同步与互斥问题。解决的办法主要有加锁法、信号量机制、管程机制等。信号量是由一个整型变量和一个等待队列构成的, 对这个整型变量除了做初始化之外, 只能实施 P-V 操作, 即 P 操作和 V 操作。P-V 操作是原子操作, 不可分割使用, 必须成对出现, 属于低级的进程通信原语。

采用 P-V 操作实现进程同步的步骤是首先为各并发进程设置私用信号量, 然后为私用信号量赋初值, 最后利用 P-V 原语和私用信号量规定各进程的执行顺序。采用 P-V 操

作实现进程互斥的步骤是首先为临界资源设置共用信号量，然后为公用信号量赋初值，最后利用 P - V 原语和公用信号量实现并发进程的互斥使用临界资源。

在本题中，收银员进程和顾客付款进程类似于生产者和消费者进程，顾客要想付款必须判断收银员是否忙，忙则等待，不忙则付款。因此，对应的收银员进程和顾客付款进程私有信号量分别是 S1 和 S2，由于开始时没有顾客去付款，收银员也没有收费，所以 S1 和 S2 的初值为 0。这里的公用临界资源是允许进入的（可容纳的）顾客人数 n，如果有 n+1 个顾客想购物，那么第 n+1 个顾客就得等待，直到有人购物完毕出来方能进去，所以临界资源的信号量为  $S_n=n$ 。

当一名顾客进入超市时，当前允许进入的顾客人数应减 1，执行 P(Sn)，购物完毕准备付款时，通知收银员进程有顾客付款，执行 V(S1)，此时收银员进程执行 P(S1)操作后就可进入收费，收费完成后收银员进程执行 V(S2)，以通知顾客收费完毕，此时顾客执行 P(S2)就可离开收银台，在离开超市时需执行 V(Sn)，释放资源。

例题答案：(22) C            (23) D            (24) C            (25) D            (26) A

**例题 2 (2002 年试题 42~46)**

一台 PC 计算机系统启动时，首先执行的是 (42)，然后加载 (43)。在设备管理中，虚拟设备的引入和实现是为了充分利用设备，提高系统效率，采用 (44) 来模拟低速设备（输入机或打印机）的工作。

已知 A、B 的值及表达式  $A^2/(5A+B)$  的求值过程，且 A、B 已赋值，该公式求值过程可用前驱图 (45) 来表示，若用 P - V 操作控制求值过程，需要 (46) 的信号量。

供选择的答案：

- (42) A. 主引导记录            B. 分区引导记录            C. BIOS 引导程序            D. 引导扇区
  - (43) A. 主引导记录和引导驱动器的分区表，并执行主引导记录
  - B. 分区引导记录、配置系统，并执行分区引导记录
  - C. 操作系统，如 Windows 98/NT/2000/XP、UNIX 等
  - D. 相关支撑软件
  - (44) A. Spooling 技术，利用磁带设备    B. Spooling 技术，利用磁盘设备
  - C. 脱机批处理系统                    D. 移臂调度和旋转调度技术，利用磁盘设备
- (45)

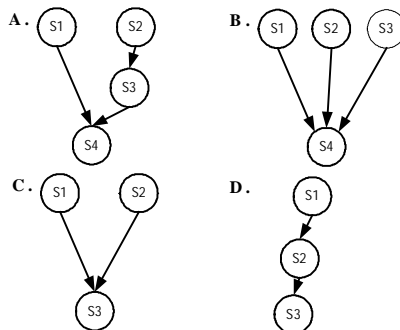


图 3-8 公式求值过程前驱图

- (46) A. 3 个且初值等于 1                      B. 2 个且初值等于 0  
       C. 2 个且初值等于 1                      D. 3 个且初值等于 0

### 例题分析:

本试题考查的是操作系统设备管理、进程管理中使用 P - V 操作描述前趋关系的知识。

#### 1. 操作系统装入过程

每当开机时自动执行 BIOS 引导程序，它主要执行如下任务。

(1) 标识和配置所有的即插即用设备：如果系统有即插即用设备，系统将搜索和测试所有安装的即插即用设备，并为它们分配 DMA 通道、IRQ 及需要的其他设备。

(2) 完成加电自检：加电自检主要检测和测试内存、端口、键盘、视频适配器、磁盘驱动器等基本设备。有一些新版本的系统还支持 CD-ROM 驱动器。

(3) 对引导驱动器可引导分区定位：在 CMOS 中，用户可以设置系统中的引导顺序，以便对引导驱动器的可引导分区重新定位。大多数系统的引导顺序是软驱，然后是硬驱，其次是 CD-ROM 驱动器。

(4) 加载主引导记录及引导驱动器的分区表，执行主引导记录 MBR。主引导记录在硬盘上找到可引导分区后，将其分区引导记录装入内存，并将控制权交给分区引导记录。由分区引导记录定位根目录，然后装入操作系统。

通过以上分析，(42) 的正确答案是 C，(43) 的正确答案是 A。

#### 2. Spooling 系统

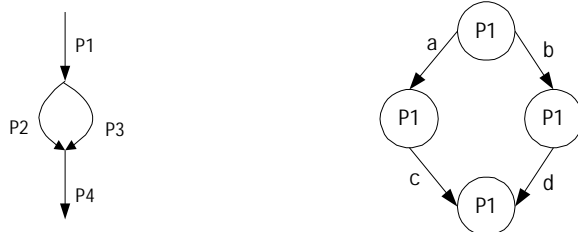
脱机的输入、输出技术是为了缓和 CPU 高速性与 I/O 设备的低速间的矛盾而引入的，详细分析参见 3.4 节的“虚设备与 Spooling 技术”相关知识点。故 (44) 答案是 B。

#### 3. 处理机管理

前驱图是一个有向无循环图，图中的每一个节点都可以表示一条语句、一个程序段或一个进程，结点的有向边表示两个节点间存在的偏序或前驱关系“→”。

其中  $\rightarrow = \{ (P_i, P_j) \mid P_i \text{ 必须在 } P_j \text{ 开始执行之前完成} \}$  如果  $(P_i, P_j) \in \rightarrow$ ，可以记为  $P_i \rightarrow P_j$ ，表示直接前驱关系，而  $P_i \rightarrow P_j \rightarrow \dots \rightarrow P_k$  表示间接前驱关系。

若干进程合作完成一个任务，这些合作进程的操作存在一定的先后次序。例如，进程 P1 ~ P4 的执行存在如图 3-9 (a) 所示的先后次序，可画出如图 3-9 (b) 所示的进程间制约关系，其中边上的字母代表信号量，边的起始端进程在执行结束前要对该信号量实施 V 操作，而边的结束端进程在执行前要对该信号量实施 P 操作，这样就确保了合作进程按约定的先后次序执行。



(a) 进程执行顺序图

(b) 进程执行前驱图

图 3-9 进程间的制约关系

因此，对于求值公式  $A^2/(5A+B)$ ，且 A、B 已赋值，该公式求值过程可用如图 3-10 所示的前驱图表示。

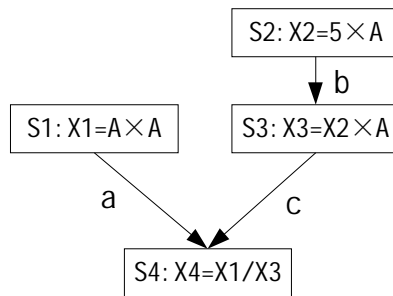


图 3-10 求值公式  $A^2/(5A+B)$  的前驱图

从图 3-10 中我们可以看出，采用 P-V 操作实现前驱图所表示的进程同步问题，共需要三个信号量（图中的 a、b、c），且信号量的初值都为 0。故试题（45）和（46）的正确答案是 A 和 D。

例题答案：（42）C           （43）A           （44）B           （45）A           （46）D

### 例题 3（2001 年试题 36~40）

（1）因争用资源产生死锁的必要条件是互斥、循环等待、不可抢占和 (36)。对于缓冲池（大量缓冲区）的管理，采用生产者—消费者方式解决同步或互斥时，通常需要用 (37) 个信号量。

（2）虚存页面调度算法有多种，(38) 调度算法不是页面调度算法。

（3）实存的存储分配算法用来决定输入的程序和数据放到主存中的位置，采用“总是把程序装入主存中最大的空闲区域”的算法称为 (39)。

（4）在文件存储设备管理中，有三类常用的空闲块管理方法，即位图向量法，空闲块链表链接法和 (40)。

供选择的答案：

（36）A. 请求与释放    B. 释放与保持    C. 释放与阻塞    D. 保持与等待

（37）A. 1            B. 2            C. 3            D. 4

（38）A. 后进先出    B. 先进先出    C. 最近最少使用    D. 随机选择

（39）A. 最佳适应算法    B. 最差适应算法    C. 首次适应算法    D. 循环首次适应算法

（40）A. 一级目录法    B. 多级目录法    C. 分区法            D. 索引法

例题分析：

第（1）小题主要考核进程管理中有关死锁和进程同步与互斥的知识点。

当多个任务竞争同样的两个或多个临界资源时，就会出现死锁。产生死锁的必要条件是互斥条件、不可抢占条件、保持与等待条件、循环等待条件。

当采用生产者—消费者方式解决同步与互斥时，通常需要两个私用信号量 empty、full 和一个公用信号量 mutex，empty 是表示空缓冲区数目的信号量，full 是表示满缓冲区数目的信号量，mutex 是对临界缓冲区进行操作的互斥信号灯。

第(2)小题主要考核存储管理中虚拟存储管理的知识点。

虚拟存储器是内存一定程度上的扩展,使得程序可以在透明的情况下访问比内存大得多的地址空间,使运行内存需要大于内存实际容量的程序成为可能,操作系统从硬盘中分配一部分空间作为虚拟存储器,在需要时同内存中的页面进行相互替换。

虚存的页面调度算法有多种,参看3.3小节的相关知识点。

第(3)小题主要考核存储管理中实存管理的知识点。

实存的可变式动态分区分配在作业执行前并不建立分区,而是在处理作业过程中按需要建立分区。关于实存管理的分配算法,参看3.3小节相关知识点。

第(4)小题主要考核文件管理中文件存储设备及存储空间管理方面的知识点。

在文件存储设备管理中,有三类常用的空闲块管理方法,即位图向量法、空闲块链表法和索引法。位图向量法中位图的每个字的每一位都对应一个物理块。空闲块链表法是把所有的空白块链在一起,当创建文件需要一块或几块时,就从链头上依次取下,反之,回收空间时,把这些空白块依次链到链头上。索引法是在文件存取器上每一个连续的空闲区建立一个索引,这种分配技术适于建立连续文件。

例题答案:(36) D (37) C (38) A (39) B (40) D

#### 例题4(2000年试题2)

当存储器采用段页式管理时,主存被划分为定长的A,程序按逻辑模块分成B。在某机器的多道程序环境下,每道程序还需要一个C作为用户标志号,每道程序都有对应的D。一个逻辑地址包括C x、段号 s、页号 p 和页内地址 d 四个部分。

设逻辑地址长度分配如下,其中 x、s、p、d 均以二进制数表示。

21	20	19	14	13	11	10	0
x		s		p		D	

其转换后的物理地址为E。

供选择的答案:

- A: ①段                      ②页                      ③区域                      ④块
- B: ①区域                      ②页                      ③块                      ④段
- C: ①模块号                      ②区域号                      ③基号                      ④区域
- D: ①一个段表和一个页表                      ②一个段表和一组页表
- ③一组段表和一个页表                      ④一组段表和一组页表
- E: ① $x*2^{20}+s*2^{14}+p*2^{11}+d$                       ② $((x+s)+p)+d$
- ③ $((x+s)+p)*2^{11}+(d)$                       ④ $((x+s)+p)*2^{11}+d$

例题分析:

本题涉及存储管理知识,主要考查段页式存储管理。

分页管理能有效地提高内存利用率,而分段系统能够反映程序的逻辑结构以满足用户的需要,并且还可以实现段的共享。将这两种存储管理方式结合起来就形成了段页式存储管理。

在段页式存储管理中，作业的地址空间按段式管理进行，即分成若干个逻辑分段，每段都有自己的段号，每段内部再分成若干大小固定的页。对内存空间的管理仍然和页式管理一样，将其分成若干个和页面大小相同的物理块，并且对内存的分配是以物理块为单位，在段页式系统中，作业的地址结构包含段号、页号及页内位移三部分，其示意图如图 3-11 所示。

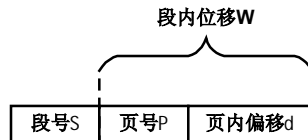


图 3-11 段页式的虚拟地址结构

系统为每个进程建立一个段表，为每个段建立一个页表。对于多道程序环境，每道程序有一个基号与其他程序相区分，每道程序可以有多个段，但只有一个段表，每个程序可以有多个页表。

段页式存储体系中逻辑地址与物理地址的转换：首先由基号段号得到段表的地址，将段内位移  $W$  的高几位划分为页表，低位则划分为页内位移。再访问段表得到页表的地址，由页表得到物理块的地址，此时得到的地址是高 11 位的地址，因此，需乘以  $2^{11}$  再加上页内地址，才得到真正的物理地址。

例题答案：A. ②      B. ④      C. ③      D. ②      E. ④

例题 5（2000 年试题 4）

  A   是操作系统中可以并行工作的基本单位，也是核心调度及资源分配的最小单位，它由   B   组成，它与程序的重要区别之一是   C  。

在 SMP 系统中，操作系统还提供了   D   机制，它是   E   的最小单位。

供选择的答案：

- |                  |             |              |               |
|------------------|-------------|--------------|---------------|
| A: ①作业           | ②过程         | ③函数          | ④进程           |
| B: ①程序、数据和标识符    | ②程序、数据和 PCB | ③程序、标识符和 PCB | ④数据、标识符和 PCB  |
| C: ①程序可占用资源，而它不可 | ②程序有状态，而它没有 | ③它有状态，而程序没有  | ④它能占有资源，而程序不能 |
| D: ①约束           | ②线程         | ③共享          | ④分时           |
| E: ①存储器分配        | ②资源分配       | ③处理器分配       | ④网络节点分配       |

例题分析：

本题考核关于进程管理中进程的概念与线程方面的知识。

进程是一个程序关于某个数据集的一次运行。也就是说，进程是运行中的程序，是程序的一次运行活动。相对于程序，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，因而进程具有动态性和并发性。

进程是一个动态的概念，在操作系统中，引入数据结构——PCB（进程控制块）来标记进程。PCB 是进程存在的惟一标志，PCB 描述了进程的基本情况。从静态的观点看，进

程由程序、数据和进程控制块组成。

对称多处理器系统 SMP 是由若干同构甚至相同的处理器构成一个系统，其操作系统提供了线程（Thread）机制以发挥多个处理器的作用在多线程系统中，一个进程可以由一个或多个线程构成。进程是资源分配的基本单位，而线程是处理器分配的最小单位。

例题答案：A. ④ B. ② C. ③ D. ② E. ③

### 例题 6 (1999 年试题 4)

进程是操作系统中的一个重要概念。进程是一个具有一定独立功能的程序在某个数据集合上的一次 A。

进程是一个 B 的概念，而程序是一个 C 的概念。

进程的最基本状态有 D。在一个单处理机中，若有 6 个用户进程，在非管态的某一时刻，处于就绪状态的用户进程最多有 E 个。

供选择的答案：

- |               |        |            |        |
|---------------|--------|------------|--------|
| A: ① 单独操作     | ② 关联操作 | ③ 运行活动     | ④ 并发活动 |
| B: ① 静态       | ② 动态   | ③ 逻辑       | ④ 物理   |
| C: ① 物理       | ② 逻辑   | ③ 动态       | ④ 静态   |
| D: ① 就绪、运行、隐蔽 |        | ② 停止、就绪、运行 |        |
| ③ 运行、就绪、阻塞    |        | ④ 就绪、撤销、运行 |        |
| E: ① 5        | ② 0    | ③ 1        | ④ 4    |

例题分析：

本题考查进程管理中进程概念及进程状态转换方面的知识，与 1995 年试题 1 基本相同。

#### (1) 进程和程序的关系

进程是一个动态的概念，而程序则是一个静态的概念。程序是指令的有序集合，没有任何执行的含义。而进程则强调执行过程，它动态地被创建，并被调度执行后消亡。举例来说，程序好比是曲谱，而进程就像是按照曲谱演奏音乐。

进程具有并发特征，而程序没有。进程的并发特征有两个方面：独立性和异步性。即并发各进程的执行是独立的，执行速度是异步的。程序不是执行过程，所以不具备并发性。

进程是操作系统中基本的并行单位、资源分配单位和调度单位。通常，进程可分为用户进程和系统进程两类，前者控制用户作业的运行，后者完成系统内部分工的管理工作。

从静态的角度看，进程由程序、数据和进程控制块（JCB）组成；从动态的角度看，进程是计算机状态的一个有序集合。

进程是一个具有一定独立功能的程序在某个数据集合上的一次运行，其中可能要涉及多个程序；而一个程序的运行过程中可能有若干进程依次或并行活动。

#### (2) 进程的状态转换

进程的基本状态有运行、就绪和阻塞三种，现代操作系统中还有挂起状态。

运行态是指进程正占用处理机。它可能因等待某个事件而阻塞，可能因时间片到而转入就绪态，也可能因其他非资源请求原因而直接挂起。

就绪态是指一个进程一切具备只等运行而所处的状态。它可能因轮到该进程占有处理的时间片或处理机空闲而转变为运行态。

阻塞态是指正等待某事件的发生而所处的状态。它可能因具备了运行的条件而转变为就绪态。注意，阻塞态不能直接进入运行态。

在一个单处理机中，处理器只有一个，非管态（即用户进程执行状态）的某一时刻，处于运行态的进程最多只有一个，系统中处于就绪态或阻塞的进程可能有多个，这样处于就绪态的进程数最多只能是进程总数减 1；如果除了运行态的一个进程外，其余进程均处于阻塞态，则就绪态进程个数为 0。

例题答案：A. ③      B. ②      C. ④      D. ③      E. ①

### 例题 7（1998 年试题 5）

信号量是操作系统中用于互斥和同步机制的一个共享的整数变量。信号量仅可以由初始化、唤醒（Signal）和等待（Wait）三种操作访问。

对于给定的信号量 S，等待操作 Wait(S)（又称 P 操作）定义为：

if  $S > 0$  then A else 挂起调用的进程

唤醒操作 Signal(S)（又称 V 操作）定义为：

if 存在等待的进程 then 唤醒这个进程 else B

给定信号量 S，可以定义一个临界区来确保其互斥，即保证在同一时刻这个临界区只能够被一个进程执行。当 S 被初始化为 1 时，代码段：

C ;  
{临界区}

D

定义了一个临界区。

这样的临界区实际上是将共享数据和对这些数据的操作一起封装起来，通过其互斥机制一次只允许一个进程进入，这种临界区通常称为 E。

供选择的答案：

- A~D: ①S:=0                      ②S:=S+1                      ③S:=S-1                      ④S:=1  
         ⑤Signal(S+1)                  ⑥Wait(S-1)                  ⑦Signal(S)                  ⑧ Wait(S)  
E: ①模块                              ②类程                              ③管程                              ④线程

例题分析：

本题主要考查进程的同步与互斥机制，以及管程的概念。

信号量是操作系统中用于互斥和同步机制的一个共享整数变量，除初始化外，仅能通过两个标准的原子操作（Atomic Operation）、Wait(S)和 Signal(S) 来访问。

对于给定的信号量 S，等待操作 Wait(S)（又称为 P 操作 P(S)）为：

Wait(S)  
{if  $S > 0$  then  $S := S - 1$  else 挂起调用的进程}

即信号量值大于 0 时，将其值减 1，进程继续，否则当前进程阻塞自己。

唤醒操作 Signal(S)（又称为 V 操作 V(S)）为：

Signal(S)  
{if 存在等待的进程 then 唤醒它 else  $S := S + 1$ ，允许其他进程访问此临界资源}

临界区（Critical Section）是指一段必须互斥执行的代码。显然，若能保证每个进程互

斥地进入自己的临界区，就能实现它们对临界资源的互斥访问。这样，每个进程进入临界区，访问该资源，并设置信号量，表示资源正在被访问，否则应等待（挂起），这个操作即 Wait(S)。当其访问完临界资源，退出临界区时，检查若有进程被挂起（即在等待访问临界资源），则唤醒该进程，否则应当恢复信号量，以使其他进程将来能访问此临界资源，这个操作即 Signal(S)。代码格式为：

```
Wait(S)
{临界区}
Signal(S)
```

如上定义的代码段中，临界区将共享数据和对这些共享数据的操作封装起来，在进入临界区时实施 Wait(S)操作和出临界区实施 Signal(S)操作，这就保证了某一时刻，只允许一个进程进入，而其他进程只能等待，这样的代码段称为管程。管程是一种并发性的构造，它包括变量、数据及过程组成的集合，只能通过特定的管程入口被调用，在边界上严格实施互斥。管程采用信息掩蔽的方法，即管程中的数据包括管程中所有过程的全局变量及某个特定过程的局部变量，所有这些数据只能在管程中访问，管程外的进程无法访问管程内的数据。

例题答案：A. ③                      B. ②                      C. ⑧                      D. ⑦                      E. ③

#### 例题 8 (1996 年试题 6)

由于内存大小有限，为使得一个或多个作业能在系统中运行，常需要外存来换取内存。其中以作业为单位对内外存进行交换的技术称为   A   技术，在作业内部对内外存进行交换的技术称为   B   技术。用外存换内存是以牺牲程序运行时间为代价的。为提高 CPU 有效利用率，避免内外存的频繁交换，  B   技术常用某种   C   来选择换出内存的页面，它的基础是程序的   D  。据此，操作系统可根据   E   来改善系统的性能，  E   是一个进程在定长的执行时间内涉及到的页面的集合。

供选择的答案：

- A、B: ① Spooling                      ② Swapping                      ③ 虚拟存储                      ④ 虚拟机  
           ⑤ 进程管理                      ⑥ 设备管理
- C:     ① 页面分配策略                ② 页面标志策略                ③ 页面淘汰策略                ④ 段设置策略
- D:     ① 完整性                          ② 局部性                          ③ 递归性                          ④ 正确性
- E:     ① 工作集                          ② 页面集                          ③ 段号集                          ④ 元素集

例题分析：

在多道程序系统中，通常内存容量远远小于各并发进程所需的空间之和。这时需要借助外存作为后备，采用信息在内、外存之间调入调出的方式，实现比内存大得多的并行进程得以运行。内外存间交换信息的基本单位有两种：一种是以作业为单位，这就是 Swapping 技术（交换技术）；另一种是在作业内部以“页”或“段”为单位进行部分信息的调入调出，这就是虚拟存储技术。

Swapping 技术可以将暂不需要的作业移到外存，让出内存空间以调入其他作业，交换到外存的作业也可以被再次调入，广泛应用于分时系统中，如早期的 UNIX 系统技术就采用 Swapping 技术。虚拟存储技术在内存中保留作业进程的一部分，而在外存放置作业进程

的副本，以页（段）为交换单位，有请求式分段、请求式分页、段页式等方式。目前如 UNIX，Windows NT，OS/2 等均采用虚拟存储技术。

但是，不管是 Swapping 技术，还是虚拟存储技术，用外存换内存是以牺牲程序运行时间为代价的。内外存之间交换的信息量越大，次数越频繁，时间越长，系统的效率就越低。为了避免内外存的频繁交换，虚拟存储技术需要选择合适页面置换算法，即页面淘汰策略。

虚拟存储管理的基础是程序的局部性原理。程序局部性有两个方面的含义：时间局部性，即如果一条指令被执行，则在不久以后可能再次执行，产生时间局限的主要原因是程序中有大量的循环操作；空间局部性，即一旦程序访问了某个内存单元，不久以后，其附近的内存单元也要被访问。

根据程序的局部性理论，Denning 提出了工作集理论。工作集是指进程运行时被频繁访问的页面集合。操作系统可以利用工作集模型来改善系统的性能，虽然程序只需少量的几页内存就可以运行，但为了使程序有效地运行，必须使程序的工作集全部在内存（主存储器）当中，否则会使进程在运行中频繁出现缺页中断，从而出现频繁的页面调入/调出现象。

例题答案：A. ②      B. ③      C. ③      D. ②      E. ①

#### 例题 9（1994 年试题 4）

从下列 10 个叙述中，选择 5 条正确的叙述： A 、 B 、 C 、 D 、 E 。

- ① 唤醒：挂起→就绪。
- ② 封锁：就绪→挂起。
- ③ 调度：就绪→运行。
- ④ 超时：运行→挂起。
- ⑤ 超时：运行→就绪。
- ⑥ 用户进程可激发调度进程。
- ⑦ 用户进程可激发唤醒进程。
- ⑧ 用户进程可激发超时进程。
- ⑨ 进程接近 CPU 可接纳的限度时，可降低页面出错的频率。
- ⑩ 进程具有引用局部性时，可降低页面出错的频率。

#### 例题分析：

进程就是系统进行分配和调度的最小单位，有挂起、就绪和运行三种基本状态。在挂起状态时不具备运行条件，等待某一事件的发生；在就绪状态时进程已分到资源，等待处理机。在运行态时进程正占有处理机。进程的状态是可变化的，其变化如图 3-12 所示。

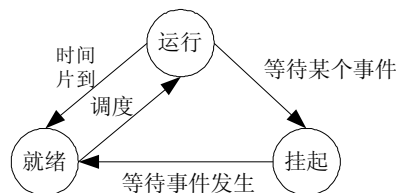


图 3-12 进程状态转换图

就绪→运行：条件是被调度程序选中。

运行→就绪：条件是时间片到（超时），或被更高优先级的进程剥夺。

运行→挂起：条件是不具备运行条件，等待某一事件的发生。

挂起→就绪：条件是等待的事件已发生，具备了运行条件。

从图中可以看出，在状态转换中不能由挂起态直接进入运行态，也不能由就绪态进入挂起态。故①、③、⑤、⑦正确。

操作系统对存储的管理主要是对内存的分配、保护和扩充。内存分配管理办法之一就是分页式管理。由于分配的内存太低，页面淘汰算法选择不当或程序存储空间跨度太大等原因，可能引起页面缺页率的增加。故⑩正确，⑨与缺页率无关。

例题答案：A. ①      B. ③      C. ⑤      D. ⑦      E. ⑩

### 例题 10（2004 年 5 月试题 22）

假设某计算机系统的内存大小为 256KB，在某一时刻内存的使用情况如图 3-13（a）所示。此时，若进程顺序请求 20KB、10KB 和 5KB 的存储空间，系统采用\_\_\_\_\_算法为进程依次分配内存出分配后的内存情况如图 3-13（b）所示。

起始地址	0KB	20KB	50KB	90KB	100KB	105KB	135KB	160KB	175KB	195KB	220KB
状态	已用	未用	已用	已用	未用	已用	未用	已用	未用	未用	已用
容量	20KB	30KB	40KB	10KB	5KB	30KB	25KB	15KB	20KB	25KB	36KB

(a)

起始地址	0KB	20KB	40KB	50KB	90KB	100KB	105KB	135KB	145KB	160KB	175KB	195KB	200KB	220KB
状态	已用	已用	未用	已用	已用	未用	已用	已用	未用	已用	未用	已用	未用	已用
容量	20KB	20KB	10KB	40KB	10KB	5KB	30KB	10KB	15KB	15KB	20KB	5KB	20KB	36KB

(b)

图 3-13 内存分配情况

供选择的答案：A. 最佳适应      B. 最差适应      C. 首次适应      D. 循环首次适应

例题分析：

实存的可变式动态分区分在作业执行前并不建立分区，而是在处理作业过程中按需要建分区。有以下几种分配算法。

(1) 首次适应法：把内存中的可用分区单独组成可用分区表或可用分区自由链，按起始地址递增的次序排列。每次按递增次序向后找。一旦找到大于或等于所要求内存长度的分区，则结束探索，从找到的分区中找出所要求内存长度分配给用户，并把剩余的部分进行合并。

(2) 循环适应法：首次适应法经常利用的是低地址空间，后面经常可能是较大的空白区，为使内存所有线性地址空间尽可能轮流使用到，每重新分配一次时，都在当前之后寻找。

(3) 最佳适应法：最佳适应算法是将输入作业放入主存中与它所需大小最接近的空白区中，将剩下的未用空间最小，该法要求空白区大小，从小到大次序组成空白区可用表或自由链。在进行分配时总是从最小的一个开始查询，因而找到了一个能满足要求的空白区便是最佳的一个。

(4) 最差适应法：分配时把一个作业程序放入主存中最不适合它的空白区，即最大的空白区（空闲区）内。

根据本题给出的两个表格，显然是最差适应法略。

例题答案：B

### 例题 11（2004 年 5 月试题 23~26）

若有一个仓库，可以存放 P1、P2 两种产品，但是每次只能存放一种产品。要求：

①  $w = P1 \text{ 的数量} - P2 \text{ 的数量}$

②  $-i < w < k$  ( $i, k$  为正整数)

若用 P-V 操作实现 P1 和 P2 产品的入库过程，至少需要 (23) 个同步信号量及 (24) 个互斥信号量，其中，同步信号量的初值分别为 (25)，互斥信号量的初值分别为 (26)。

- 供选择的答案：
- |           |              |            |                   |
|-----------|--------------|------------|-------------------|
| (23) A. 0 | B. 1         | C. 2       | D. 3              |
| (24) A. 0 | B. 1         | C. 2       | D. 3              |
| (25) A. 0 | B. $i, k, 0$ | C. $i, k$  | D. $i - 1, k - 1$ |
| (26) A. 1 | B. 1, 1      | C. 1, 1, 1 | D. $i, k$         |

#### 例题分析：

同步是指进程间共同完成一项任务时直接发生相互作用的关系，即具有伙伴关系的进程在执行时间次序上必须遵循的规律。互斥是指进程因竞争同一资源而相互制约。

同步和互斥可以这样来理解，互斥是指在使用临界资源的时候，多个进程不能同时使用临界资源，如果进程 A 在使用，B 需要等待，待 A 用完之后，才能让 B 用。这种信号量的初值一般为 1，表示只有一个资源可用，如果已经有一个进程占用了这个资源，其他进程要使用，则须等待。同步是指进程间共同完成一项任务时直接发生相互作用的关系，即具有伙伴关系的进程在执行时间次序上必须遵循的规律。通俗一点说就是要步伐一致，即保证差距不是很远。例如 A 和 B，两人约定去 C 家里玩，不过 A 开车，B 骑自行车，他们的速度肯定是不一样的，他们同时出发，车开一段距离，就停下来等 B，当 B 快追上 A 时，A 再开始前进，这就是一个同步的过程。

同步和互斥的思想引入到存货取货的进程 A、B 中，A 存货，B 取货。他们在同一个货仓作业，所以不能同时工作。要么 A 存货，要么 B 取货。这样他们之间就要设定一个互斥信号量，信号量初值为 1。假设货仓能存 n 件货物，我们光控制不让 A、B 同时在货仓工作是不够的，还要进行同步控制，当货仓装了 N 件货物之后，A 就不能再向货仓存货了，而要等到 B 取出一些货，才能再存入货物，这就是同步过程，是 A 在等 B 完成他的工作。

这一题最大的难点，是如何把题目中给出的两个式子，转化成为我们能够用上的条件。

题目中有说明：

1.  $w = P1 \text{ 的数量} - P2 \text{ 的数量}$ 。

2.  $-i < w < k$

这样看条件很抽象，我们要把他们转化一下就清楚了。

P1 的数量 - P2 的数量  $< k$

P2 的数量 - P1 的数量  $< i$

这也就是说：如果我们先不存 P2，则 P1 的数量  $0 < k$ ，也就是说最多 P1 可以存  $k-1$  个，就不能存了，要等到有 P2 产品存入时，才能再次存入 P1 产品；同样，如果我们先不存 P1，则 P2 的数量  $0 < i$ ，也就是说最多 P2 可以存  $i-1$  个，就不能存了，要等到有 P1 产品存入时，才能再次存入 P2 产品。有了这个关系，我们的题目也就得出答案了，需要一个互斥信号量和两个同步信号量，互斥信号量的初值为 1，而同步信号量的初值为： $k-1, i-1$ 。

例题答案：(23) C                      (24) B                      (25) D                      (26) A

### 例题 12 (2004 年 11 月试题 23~24)

虚拟存储管理系统的基础是程序的 (23) 理论，这个理论的基本含义是指程序执行是往往会不均匀的访问主存储器单元。根据这个理论，Denning 提出了工作集理论。工作集是进程运行时被频繁访问的页面集合。在进程运行时，如果它的工作集页面都在 (24) 内，能够使该进程有效的运行，否则会出现频繁的页面调入/调出现象

供选择的答案：

- (23) A.全局性                      B.局部性                      C.时间全局性                      D.空间全局性  
 (24) A.主存储器                      B.虚拟存储器                      C.辅助存储器                      D.U 盘

试题分析：

与 1990 年试题 2，1998 年试题 7 基本相同。虚拟存储管理系统的基础是程序的局部性理论。所谓程序局部性原理是指程序在执行时所呈现的局部性规律，即在一段较短时间内，程序的执行仅限于某个部分。局部性原理又是表现为两个方面。

- I 时间局限性：如果某条指令被执行，则不久以后该指令很可能再次被执行；如果某条数据结构被访问，则不久以后该数据结构很可能再次被访问。产生时间局限性的主要原因是程序中有大量的循环操作。
- I 空间局限性：一旦程序访问了某个内存单元，不久以后，其附近的内存单元也要被访问，即程序在一段时间内所访问的存储器空间可能集中在一定的范围之内，其最常见情况就是程序的顺序执行。

工作集是指在进程运行时被频繁访问的页面集合。虽然程序只需少量的几页内存就可以运行，但为了使程序更有效地运行，必须使程序的工作集全部在内存（主存储器）当中，否则会使进程在运行中频繁出现缺页中断，从而出现频繁的页面调入/调出现象。

试题答案：(23) B                      (24) A

### 例题 13 (2004 年 11 月试题 27)

假设系统中有三类互斥资源 R1、R2、R3，可用资源数分别是 9、8、5。在 T0 时刻系统中有 P1、P2、P3、P4 和 P5 五个进程，这些进程对资源的最大需求量和已分配资源数如表 3-2 所示，如果进程按 (27) 序列执行，那么系统状态是安全的。

表 3-2 进程资源表

进程 \ 资源	最大需求量			已分配资源数		
	R1	R2	R3	R1	R2	R3
P1	6	5	2	1	2	1
P2	2	2	1	2	1	1
P3	8	0	1	2	1	0
P4	1	2	1	1	2	0
P5	3	4	4	1	1	3

供选择的答案：

(27) A.P1->P2->P4->P5->P3

B.P2->P1->P4->P5->P3

C.P2->P4->P5->P1->P3

D.P4->P2->P4->P1->P3

试题分析：

安全状态，是指系统能按某种进程顺序（P1，P2，…，Pn），来为每个进程 Pi 分配其所需的资源，直到满足每个进程对资源的最大需求，使每个进程都可以顺利完成。如果无法找到这样的一个安全序列，则称系统处于不安全状态。

本题，序列已经给出，我们只需将四个选项按其顺序执行一遍，便可以判断出现死锁的三个序列。

首先求，剩下的资源数为：

$$R1=9-(1+2+2+1+1)=2;$$

$$R2=8-(2+1+1+2+1)=1;$$

$$R3=5-(1+1+3)=0.$$

由于 R3 资源为 0，系统不能在分配 R3 资源了，所以不能一开始就运行需要分配 R3 资源的进程。所以，A 和 D 显然是不安全的。

其次，求序列 P2->P4->P5->P1->P3 是否安全，进程运行安全情况分析如表 3-3 所示。

表 3-3 进程运行分析表 1

进程 \ 资源	Work			Need			Allocation			Work+Allocation			Finish
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
P2	2	1	0	0	1	0	2	1	1	4	2	1	True
P4	4	2	1	0	0	1	1	2	0	5	4	1	True
P5	5	4	1	2	3	1	1	1	3	6	5	4	True
P1	6	5	4	5	3	1	1	2	1	7	7	5	True
P3	7	7	5	6	0	1	2	1	0	9	8	5	True

显然，该序列是安全的。

最后，求序列 P2->P4->P5->P1->P3 是否安全，如表 3-4 所示。

表 3-4 进程运行分析表 2

进程 \ 资源	Work			Need			Allocation			Work+Allocation			Finish
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
P2	2	1	0	0	1	0	2	1	1	4	2	1	True
P1	4	2	1	5	3	1	1	2	1				

这时候，我们发现进程 P1 需要 R1 资源为 5，我们能提供的 R1 资源为 4，所以序列无法进行下去，为不安全序列。

试题答案：C

例题 14 (2005 年 5 月试题 26~27)

在一个单 CPU 的计算机系统中，有两台外部设备 R1、R2 和三个进程 P1、P2、P3。系统采用可剥夺方式优先级的进程调度方案，且所有进程可以并行使用 I/O 设备，三个进程的优先级、使用设备的先后顺序和占用设备时间如表 3-5 所示。

表 3-5 设备的先后顺序和占用设备时间

进 程	优 先 级	使用设备的先后顺序和占用设备时间
P1	高	R2(30ms)→CPU(10ms) →R1(30ms) →CPU(10ms)
P2	中	R1(20ms)→CPU(30ms) →R2(40ms)
P3	低	CPU(40ms) →R1(10ms)

假设操作系统的开销忽略不计，三个进程从投入运行到全部完成，CPU 的利用率约为 (26) %；R2 的利用率约为 (27) % (设备的利用率指该设备的使用时间与进程组全部完成所占用时间的比率)。

供选择的答案：

(26) A.60      B.67      C.78      D.90

(27) A.70      B.78      C.80      D.89

试题分析：

根据题目的描述，我们可以把系统运行的时空图画出，如图 3-14 所示。

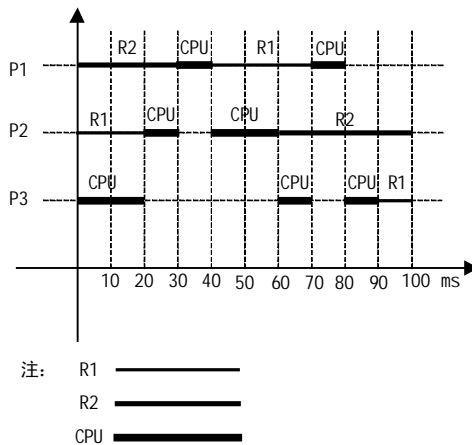


图 3-14 时空图

那么这个图是怎么来的呢，我来详细的介绍一下：首先 P1 进程使用 R2 资源 30ms，所以 p1 的前 30ms 注明为 R2。与此同时 P2 使用 R1 资源 20ms，P3 同时申请使用 CPU40ms，当 P3 申请使用 CPU 时，没其他进程申请使用 CPU，所以 P3 顺利得到了 CPU 的使用权。但我们可以看到，当系统时间到 20ms 时，P3 失去了 CPU 资源，这是为什么呢？因为此时 P2 已经使用完 R1，它开始申请使用 CPU 了，P2 的优先级比 P3 高，所以系统从 P3 手中收回 CPU 的使用权，把 CPU 分配给 P2 使用，当系统时间到 30ms 时，系统又从 P2 手中收回了 CPU 的使用权，把 CPU 分配给了 P1 使用，因为系统中 P1 的优先级比 P2 高。依次类推，便完成了系统时空图。

从图 3-10 我们可以看出三个进程运行完毕需要 100ms，CPU 工作了 90ms，所以 CPU 的利用率为 90%，R2 工作了 70ms 所以，R2 的利用率为 70%。

试题答案：(26) D (27) A

**例题 15 (2005 年 11 月试题 10~13)**

如图 3-16 所示的树型文件系统中，方框表示目录，圆圈表示文件，“/”表示路径的分隔符，“/”路径之首表示根目录。图 3-15 中，(10)。

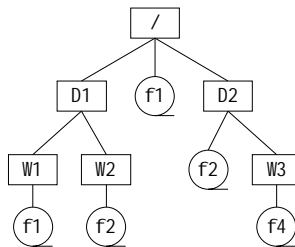


图 3-15 树型文件系统示意图

假设当前目录是 D1，进程 A 以如下两种方式打开文件 f1：

方式 1 `fd1=open("____(11)____/f1",o_RDONLY);`

方式 2 `fd1=open("/D1/W1/f1",o_RDONLY);`

其中，方式 1 的工作效率比方式 2 的工作效率高，因为采用方式 1 的文件系统 (12)。

供选择的答案：

- (10) A. 子目录 W2 中文件 f2 和子目录 D2 中文件 f2 是完全相同的
- B. 子目录 W2 中文件 f2 和子目录 D2 中文件 f2 是不相同的
- C. 子目录 W2 中文件 f2 和子目录 D2 中文件 f2 是可能相同也可能不相同
- D. 树型文件系统中不允许出现相同名字的文件
- (11) A. /D1/W1      B. D1/W1      C. W1      D. f1
- (12) A. 可以直接访问根目录下的文件 f1
- B. 可用从当前路径开始查找需要访问的文件 f1
- C. 只需要访问一次磁盘，就可以读取文件 f1，而方式 2 需要两次
- D. 只需要访问一次磁盘，就可以读取文件 f1，而方式 2 需要三次

**例题分析:**

在树型目录结构中，树的根结点为根目录，数据文件作为树叶，其他所有目录均作为树的结点。从根结点出发到达任意文件叶子结点得到唯一路径（在 Unix 中，各结点间用“/”隔开），这便是该文件的绝对路径，比如图 3-15 中左边 f1 的绝对路径为/D1/W1/f1。文件名相同，但绝对路径不同，可能是不同的文件。从当前目录开始（不包括当前目录）到达文件叶子结点的路径称为该文件的相对路径，比如若当前目录为 D1，那么图 3-15 中左边 f1 的相对路径为 W1/f1。

访问一个文件时，可采用相对路径名也可以采用绝对路径名，但前者访问目录文件的次数比后者要少。该题中，当前路径为 D1，如采用相对路径为 W1/f1 来访问 f1，只需要访问磁盘 2 次就可以读取文件 f1，第一次是在当前目录 D1 中找到 W1，第二次是在 W1 文件目录中找到文件 f1 的物理位置，接着就可以读取文件 f1 了。

**例题答案:** (10) C (11) C (12) B

**例题 16 (2005 年 5 月试题 15~16)**

数据存储于磁盘上的排列方式会影响 I/O 服务的总时间。假设每磁道划分成 10 个物理块，每块存放 1 个逻辑记录。逻辑记录 R1, R2, …, R10 存放在同一个磁道上，记录的安排顺序如表 3-6 所示。

表 3-6 记录的安排顺序

物理块	1	2	3	4	5	6	7	8	9	10
逻辑记录	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10

假定磁盘的旋转速度为 20ms/周，磁头当前处在 R1 的开始处。若系统顺序处理这些记录，使用单缓冲区，每个记录处理时间为 4ms，则处理这 10 个记录的最长时间为\_\_(15)\_\_; 若对信息存储进行优化分布后，处理 10 个记录的最少时间为\_\_(16)\_\_。

**供选择的答案:**

- (15) A.180ms    B.200ms    C.204ms    D.220ms  
 (16) A.40ms    B.60ms    C.100ms    D.160ms

**试题分析:**

首先从磁盘的转速 20ms/圈，我们可以知道，读取一条记录需要 2ms。值得注意的一点是，处理一条记录的前提，是将其读出来。所以处理第一条记录时，要先将其读取出来，再进行处理，所以处理 R1 所需时间为 2ms+4ms，当 R1 处理完时，磁头已经转到了 R4 的位置，此时要将其调整到 R2 的位置，需要经过 R5, R6, R7, R8, R9, R10, R1，这样要耗 16ms 的时间，再加上读取 R2 需要 2ms 以及处理数据的 4ms，R2 的总处理时间应为 22ms。所以  $2+4+(16+2+4)*9=204ms$ 。而优化后的排列顺序应为 R1, R8, R5, R2, R9, R6, R3, R10, R7, R4，这样的排列顺序刚好是处理完 R1，磁头就到了 R2 的位置，直接读取 R2，处理 R2，处理完 R2，磁头又到了 R3 的位置，依此类推，每条记录的读取及处理时间为  $2ms+4ms=6ms$ ，所以总时间为： $(2+4)*10=60ms$ 。

**试题答案:** (15) C (16) B

例题 17（2005 年 11 月试题 25~26）

某仓库有两名发货员，一名审核员。当顾客提货时，只要发货员空闲，允许顾客进入仓库提货，顾客离开时，审核员检验顾客提货是否正确。其工作流程如图 3-16 所示。为了利用 PV 操作正确地协调他们之间的工作，设置了两个信号量 S1 和 S2，且 S1 的初值为 2，S2 的初值为 1。图中的 a 应填写\_\_\_\_（25）\_\_\_\_；图中的 b、c 和 d 应分别填写\_\_\_\_（26）\_\_\_\_。

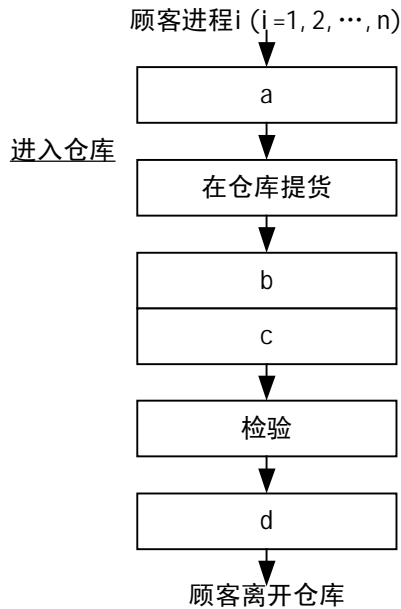


图 3-17 工作流程

供选择的答案：

- (25) A.P(S1)            B.P(S2)            C.V(S1)            D.V(S2)  
(26) A.P(S2)、V(S2)和V(S1)    B.P(S1)、V(S1)和V(S2)  
      C.V(S1)、P(S2)和V(S2)    D.V(S2)、P(S1)和V(S1)

试题分析：

首先要理解好 P、V 操作的含义。假设信号量为 sem，则：

P 原语的主要操作是：

- (1) sem 减 1；
- (2) 若 sem 减 1 后仍大于或等于零，则该进程继续执行；
- (3) 若 sem 减 1 后小于零，则该进程被阻塞，在相应队列中排队，然后转向系统的进程调度。

V 原语的主要操作是：

- (1) sem 加 1；
- (2) 若相加结果大于零，则进程继续执行；
- (3) 若相加结果小于或等于零，则唤醒一阻塞在该信号量上的进程，然后再返回原进程继续执行或转进程调度。

当信号量  $S$  小于 0 时, 其绝对值表示系统中因请求该类资源未被满足而被阻塞的进程数目,  $S$  大于 0 时表示可用的临界资源数。

对于  $V$  操作有一些说明, 以纠正理解偏差。

$Sem$  大于 0 的确表示有临界资源可供使用, 而且这个时候没有进程被阻塞在这个资源上, 也就是说没有进程因为得不到这类资源而阻塞, 所以没有被阻塞的进程, 自然不需要唤醒。有人可能会问, 当  $Sem$  小于 0 时表明没有临界资源可供使用, 为什么还要唤醒进程?

这是因为,  $V$  原语操作的本质在于一个进程使用完临界资源后, 释放临界资源, 使  $Sem$  加 1, 以通知其他的进程, 这个时候如果  $Sem < 0$ , 表明有进程阻塞在该类资源上, 因此要从阻塞队列里唤醒一个进程来“转手”该类资源。比如, 有 2 个某类资源, 三个进程 A、B、C、D 要用该类资源, 最开始  $Sem=2$ , 当 A 进入,  $Sem=1$ , 当 B 进入  $Sem=0$ , 表明该类资源刚好用完, 当 C 进入时  $Sem=-1$ , 表明有一个进程被阻塞了, D 进入,  $Sem=-2$ 。当 A 用完该类资源时, 进行  $V$  操作,  $Sem=-1$ , 释放该类资源, 而这时  $Sem < 0$ , 表明有进程阻塞在该类资源上, 于是唤醒一个。

有了上述对  $PV$  操作的正确理解, 那这道题就很好解决了。

$S1$  的初值为 2, 显然表明最开始有两个“发货员”这种资源, 当顾客去提货时要用去一个这样的资源, 于是 a 显然填  $P(S1)$ 。当提货完了之后, 顾客进程要释放“发货员”资源, 于是 b 显然填  $V(S1)$ 。从图中可看出, 接着审核员要审核提货是否正确, 同理, 顾客要用去一个“审核员”资源, 于是 c 应该填  $P(S2)$ 。最后, d 显然填  $V(S2)$  了。

值得一提的是, 很多考生记不清是  $P$  操作加 1 还是  $V$  操作加 1, 这里给大家提供一个小窍门, 大家看字母“ $V$ ”, 从下往上看其水平宽度是逐渐变大的, 这“表明” $V$  操作是对信号量进行加 1 操作。

试题答案: (25) A (26) C

### 例题 18 (2006 年 5 月试题 21~22)

为了解决进程间的同步和互斥问题, 通常采用一种称为\_\_\_\_(21)\_\_\_\_机制的方法。若系统中有 5 个进程共享若干个资源  $R$ , 每个进程都需要 4 个资源  $R$ , 那么使系统不发生死锁的资源  $R$  的最少数目是\_\_\_\_(22)\_\_\_\_。

(21) A. 调度 B. 信号量 C. 分派 D. 通讯

(22) A. 20 B. 18 C. 16 D. 15

例题分析:

为了解决进程间的同步和互斥问题, 通常采用一种称为信号量机制的方法。若系统中有 5 个进程共享若干个资源  $R$ , 每个进程都需要 4 个资源  $R$ , 那么使系统不发生死锁的资源  $R$  的最少数目是 16 个。因为如果系统有 16 个资源, 可以给每个进程先分配 3 个资源。此时还余下 1 个资源, 这个资源无论分配给哪个进程, 都能完成该进程的运行, 当此进程运行完毕可以将其所有资源释放, 所以这样系统不可能产生死锁。所以此题选 C。

例题答案: (21) B (22) C