

## 操作系统的功能、类型和层次结构



### 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：

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

批处理操作系统 ( Batch Processing Operating System )

分时操作系统 ( Time Share Operating System )

实时操作系统 ( Rea Time Operating System )

网络操作系统 ( Network Operating System )

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。

## 处理机管理（进程管理）

### 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是进程存在的唯一标志，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。

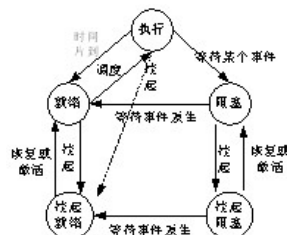


图3-3 进程状态转换

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。新建立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，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Buffu,它们的初值分别是1和0,相应的程序段形式是：

```
生产者  
oop  
生产一产品next;  
P ( Bufempty ) ;  
next产品存缓冲区；  
V ( Buffu ) ;  
endoop  
消费者  
oop  
P ( Buffu ) ;  
从缓冲区中取产品；  
V ( Bufempty ) ;  
使用产品  
endoop
```

## 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：

先来先服务（FCFS）调度算法，又称先进先出（FIFO）。就绪队列按先后来后到原则排队。

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。

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。

版权方授权希赛网发布，侵权必究

[上一节](#)    [本书简介](#)    [下一节](#)

第3章：操作系统基础知识

作者：希赛教育软考学院 来源：希赛网 2014年01月26日

## 存储管理

### 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。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4种存储分配算法介绍如下。

**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。

**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。

**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，因而找到的一个能满足要求的空白区便是最佳的一个。

**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。

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。

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。

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。

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。

方法	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	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，动态链接机构	同左

表3-1 存储管理方式比较表

版权方授权希赛网发布，侵权必究

上一节    本书简介    下一节

## 设备管理

### 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Spooing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。

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。

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。

实现设备和设备、设备和CPU等之间的并行操作。

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。

程序控制方式。

中断方式。

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。

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。

先请求先分配。

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3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。

先来先服务（FCFS）调度。按先来后到次序服务，未做优化。

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）。

SCAN调度。又称电梯算法，SCAN算法是磁头前进方向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Spooing技术

Spooing ( simultaneous periphera operation on ine )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。Spooing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Spooing,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。Spooing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。

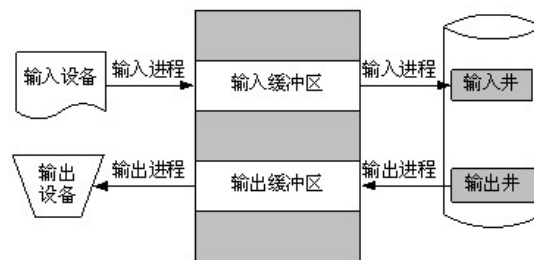


图3-5 Spooing系统示意图

版权方授权希赛网发布，侵权必究

[上一节](#)    [本书简介](#)    [下一节](#)

## 文件管理

### 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：

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使用信息。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，是文件存在的唯一标志，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4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。

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。

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。

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。

版权方授权希赛网发布，侵权必究

[上一节](#)

[本书简介](#)

[下一节](#)

## 作业管理

### 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4个状态。其状态转换如图3-6所示。

**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。

**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）。

**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。

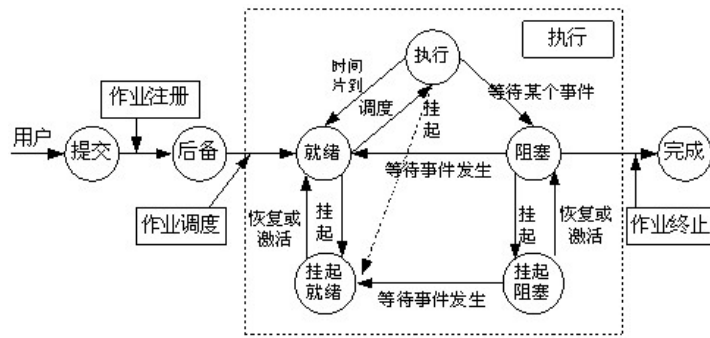


图3-6 作业的状态及其转换

完成状态：当作业正常运行结束，它所占用的资源尚未全部被系统回收时的状态为完成状态。

## 2) 处理机调度

处理机调度通常分为三级调度，即低级调度、中级调度和高级调度。

高级调度：高级调度也称为作业调度。高级调度的主要功能是在批处理作业的后备作业队列中选择一个或者一组作业，为它们建立进程，分配必要的资源，使它们能够运行起来。

中级调度：中级调度也称为交换调度，中级调度决定进程在内、外存之间的调入、调出。其主要功能是在内存资源不足时将某些处于等待状态或就绪状态的进程调出内存，腾出空间后，再将外存上的就绪进程调入内存。

低级调度：低级调度也称为进程调度，低级调度的主要功能是确定处理器在就绪进程间的分配。

## 3) 作业控制块 (JCB)

在作业管理中，系统为每一个作业建立一个作业控制块JCB.系统通过JCB感知作业的存在。JCB包括的主要内容有，作业名、作业状态、资源要求、作业控制方式、作业类型，以及作业优先权等。

### 2.作业调度及其常用调度算法

作业调度主要完成从后备状态到执行状态的转变，以及从执行状态到完成状态的转变。作业调度算法有以下几种。

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

优先级调度。根据作业的优先级别，优先级高者先调度。

## 3.用户接口

用户接口也称为用户界面，其含义有两种：一种是指用户与操作系统交互的途径和通道，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。

命令接口：命令接口包含键盘命令和作业控制命令；

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唯一接口。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。

版权方授权希赛网发布，侵权必究

[上一节](#)    [本书简介](#)    [下一节](#)

第3章：操作系统基础知识

作者：希赛教育软考学院    来源：希赛网    2014年01月26日

## 网络操作系统

### 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She、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3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She命令解释层、实用程序、库函数等，该层中的She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3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；Cose,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；ink,建立连接；Unink,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。

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；Signa,控制进程对特别事件的响应；Ki,发送软中断信号；Msgsnd,发送消息；Msgrcv,接收消息。

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。

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RU（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) She

UNIX系统中的命令解释程序称做She.命令行的形式是：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。She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。

She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s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：s>Outfie,则将s的输出重定向到当前目录下的文件Outfie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：

```
s > fie
sort < fie > fie2
pr < fie2
```

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，She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位标准字符集的单一代码方式（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。

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)    [本书简介](#)    [下一节](#)

第3章：操作系统基础知识

作者：希赛教育软考学院    来源：希赛网    2014年01月26日

## 嵌入式操作系统

---

### 3.8 嵌入式操作系统

本节将介绍嵌入式操作系统。

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)    [本书简介](#)    [下一节](#)

第3章：操作系统基础知识

作者：希赛教育软考学院    来源：希赛网    2014年01月26日

## 嵌入式操作系统概述

---

### 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，无须人工干预。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程序的运行，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。

可装卸性。开放性、可伸缩性的体系结构。

强实时性。EOS实时性一般较强，可用于各种设备控制当中。

统一的接口。提供各种设备驱动接口。

操作方便、简单，提供友好的图形界面，追求易学易用。

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。

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。

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。

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Pam OS,Microsoft公司的Windows CE和开放源代码的linux。

版权方授权希赛网发布，侵权必究

[上一节](#)    [本书简介](#)    [下一节](#)

## 常用的嵌入式操作系统

### 3.8.2 常用的嵌入式操作系统

#### 1.Pam OS

Pam OS是一种32位的嵌入式操作系统。Pam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。Pam OS是一套具有强开放性的系统，现在有大数千种专门用Pam 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。

Pam OS是一套专门为掌上电脑开发的OS.在编写程序时，Pam 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Pam OS编写的应用程序占用的空间也非常小，所以，基于Pam 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Pam产品的最大特点是使用简便、机体轻巧，因此决定了Pam OS应具有以下特点。

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Pam 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。

合理的内存管理。Pam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。

Pam 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，Pam 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。

Pam 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Visua Basic、Visua 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。

具有灵活的电源管理功能，包括睡眠/唤醒模式。

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。

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。

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。

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，并通过增强的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。

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。

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.inux

inux是一个类似于UNIX的操作系统,inux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inux OS迅速发展的状况下,inux OS逐渐形成了可与Windows CE等EOS进行抗衡的局面。嵌入式inux OS的特点如下。

精简的内核,性能高、稳定,多任务。

适用于不同的CPU,支持多种体系结构,如X86、ARM、MIPS、APHA、SPARC等。

能够提供完善的嵌入式GUI及嵌入式X-Windows.

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。

提供完整的开发工具和SDK,同时提供PC上的开发版本。

用户可定制,可提供图形化的定制和配置工具。

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。

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。

完善的中文支持,强大的技术支持,完整的文档。

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。

#### 4.3种嵌入式操作系统的比较

嵌入式inux OS与Windows CE相比的优点如下。

inux是开放源代码的,不存在黑箱技术,遍布全球的众多inux爱好者都是inux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。

inux的源代码随处可得,注释丰富,文档齐全,易于解决各种问题。

inux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。

inux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。

inux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inux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inux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。

inux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网络,还有无线网络、Token ring(令牌环)和光纤甚至卫星的支持。

inux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inux OS与Windows CE相比的弱点如下。

开发难度较高,需要很高的技术实力。

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。

与某些商业OS一样,嵌入式inux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,但是如果不小心,将引起新的问题。

有些inux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inux应用程序都可以在嵌入式系统中运行。

3Com公司的Pam 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，Pam 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。

Pam OS是一套具有极强开放性的系统。开发者向用户免费提供Pam OS的开发工具，允许用户利用该工具在Pam 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Pam OS与Windows CE的另一个主要区别。

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

版权方授权希赛网发布，侵权必究

[上一节](#)    [本书简介](#)    [下一节](#)

第3章：操作系统基础知识

作者：希赛教育软考学院    来源：希赛网    2014年01月26日

## 例题分析

### 3.9 例题分析

#### 例题1（2011年5月试题23~24）

某文件管理系统在磁盘上建立了位示图（bitmap），记录磁盘的使用情况。若系统的字长为32位，磁盘上的物理块依次编号为：0、1、2、...，那么4096号物理块的使用情况在位示图中的第（23）个字中描述；若磁盘的容量为200GB,物理块的大小为1MB,那么位示图的大小为（24）个字。

(23) A.129    B.257    C.513    D.1025

(24) A.600    B.1200    C.3200    D.6400

#### 例题分析：

位示图法是为管理磁盘空闲存储空间而提出的一种方法，该方法是在外存上建立一张位示图来记录文件存储器的使用情况。每一位仅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。

在本题中，题目告诉我们字长32位，即一个字可以表示32个物理块得使用情况，而物理块依次编号为：0、1、2、...，那么第4096号物理块的使用情况应该在位示图中的第129个字中描述，因为 $(4096+1)/32$ 等于128余1,所以这一位应属于第129个字的第一位，其中4096+1的这个1是因为编号是从0开始计数的，所以才加1。

由于磁盘的容量为200GB,每个物理块的大小为1MB,那么总共有 $200\text{GB}/1\text{MB}=200 \times 1024$ 个物理块，需要 $200 \times 1024$ 位来表示，因此位示图中字的个数= $200 \times 1024/32=6400$ 个。

例题答案：(23) A (24) D

#### 例题2（2011年5月试题25~26）

系统中有R类资源m个，现有n个进程互斥使用。若每个进程对R资源的最大需求为w,那么当m、n、w分别取表3-2中的值时，对于表中的①~⑥种情况，（25）可能会发生死锁。若将这些情况的

m分别加上 ( 26 ) ,则系统不会发生死锁。

	①	②	③	④	⑤	⑥
m	3	3	5	5	6	6
n	2	3	2	3	3	4
w	2	2	3	3	3	2

表3-2 资源分配表

( 25 ) A.①②⑤    B.③④⑤    C.②④⑤    D.②④⑥

( 26 ) A.1、1和1    B.1、1和2    C.1、1和3    D.1、2和1

**例题分析：**

在本题中，第①种情况的系统资源数为3,而有2个进程互斥使用这3个资源，每个进程需要的资源最大值为2,那么无论怎么分配资源，都不会发生死锁。第②种情况的系统资源数为3,有3个进程互斥使用这3个资源，每个进程需要的资源最大值为2,如果每个进程都分配一个该类资源而又互相等待，这时就有可能产生死锁。第③种情况的系统资源数为5,而有2个进程互斥使用这5个资源，每个进程需要的资源最大值为3,那么无论怎么分配资源，都不会发生死锁。第④种情况的系统资源数为5,有3个进程互斥使用这3个资源，每个进程需要的资源最大值为3,如果有两个进程分配了两个资源，而剩余的一个该类资源分配给了第3个进程，这时就有可能产生死锁。对于第⑤种情况，如果三个进程都分配了2个资源而互相等待则会产生死锁。

经过上面的分析我们可以知道，②④⑤可能会产生死锁，对于第②种情况，由于每个进程都分配一个该类资源，如果系统再多一个该类资源，系统将不会产生死锁。对于第④种情况，由于有两个进程分配了两个资源，而第3个进程只分配了1个资源，如果系统只增加一个资源的话，那么将增加的这个资源分配给第3个进程，这个时候系统仍然会发生死锁，只有增加两个资源系统才不会死锁。第⑤中情况与第②种情况一样，只需要增加一个资源就不会发生死锁。

**例题答案：( 25 ) C ( 26 ) D**

**例题3 ( 2011年5月试题27~28 )**

某系统采用请求页式存储管理方案，假设某进程有6个页面，系统给该进程分配了4个存储块，其页面变换表如表3-3所示，表中的状态位等于1/0分别表示页面在内存/不在内存。当该进程访问的页面2不在内存时，应该淘汰表中页号为 ( 27 ) 的页面。假定页面大小为4K,逻辑地址为十六进制3C18H,该地址经过变换后的页帧号为 ( 28 )。

页号	页帧号	状态位	访问位	修改位
0	5	1	1	1
1	-	0	0	0
2	-	0	0	0
3	2	1	1	0
4	8	1	1	1
5	12	1	0	0

表3-3 页面变换表

( 27 ) A.0    B.3    C.4    D.5

( 28 ) A.2    B.5    C.8    D.12

**例题分析：**

本题主要考查页式存储管理。

在分页存储管理时，将内存划分为大小相等的页面，每一页物理内存叫页帧，以页为单位对内存进行编号，该编号可作为页数组的索引，又称为页帧号。在淘汰页面时，应选择页帧号最大的进

行淘汰，因此当该进程访问的页面2不在内存时，应该淘汰页号为5的页面。

另外，题目高位我们页面大小为4K,即需要12位来表示其存储空间，而逻辑地址3C18H转换为二进制为0011 1100 0001 1000,其低12位为页内地址，而高4位为页号，即0011,,转换为十进制后结果为3,查表可知，页号为3的页面对应的页帧号为2。

**例题答案：( 27 ) D ( 28 ) A**

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)    [本书简介](#)    [下一节](#)

第 4 章：软件工程基础知识

作者：希赛教育软考学院 来源：希赛网 2014年01月26日

## 软件生命周期与软件开发模型

---

### 第4章 软件工程基础知识

根据考试大纲，要求考生掌握软件生命周期各阶段的任务、结构化分析和设计方法、面向对象的分析与设计、软件开发工具与环境的基础知识、软件质量保证的基础知识、软件过程改进与评估和软件项目管理基础知识等七个方面的知识。

#### 4.1 软件生命周期与软件开发模型

本节将介绍软件生命周期与软件开发模型。

[版权方授权希赛网发布，侵权必究](#)

[上一节](#)    [本书简介](#)    [下一节](#)

第 4 章：软件工程基础知识

作者：希赛教育软考学院 来源：希赛网 2014年01月26日

## 软件危机与软件工程

---

### 4.1.1 软件危机与软件工程

软件工程是一门年轻的学科。"软件工程"这个概念最早是在1968年召开的一个当时被称为"软件危机"的会议上提出的。自1968年以来，我们在该领域已经取得了长足的进步。软件工程的发展已经极大地完善了我们的软件，使我们对软件开发活动也有了更深的理解。

#### 1. 软件危机

那么什么是软件危机呢？自从强大的第三代计算机硬件问世以后，许多原来难以实现的计算机应用成为现实，同时对软件系统的需求数量和复杂度要求变得更高。而当时的软件开发技术无法满足这一日益增长的需求，引发了软件危机。它主要表现为。

软件开发生产率提高的速度，远远跟不上计算机迅速普及的趋势。软件需求的增长得不到满足，软件产品"供不应求"的现象使人类无法充分利用现代计算机硬件提供的巨大潜力。

软件成本在计算机系统总成本中所占的比例逐年上升。

不能正确估计软件开发产品的成本和进度，致使实际开发成本高出预算很多，而且超出预期的开发时间要求。