

第一章

- 1、设计现代 OS 的主要目标是：有效性、方便性、可扩展性、开放性。
- 2、操作系统的五大功能是：处理机管理、存储器管理、设备管理、文件管理、用户管理。
- 3、操作系统的基本特性是：并发性、共享性、异步性、虚拟性。其中最基本特征是并发和共享。最重要的特征是并发性
- 4、操作系统的作用：作为用户与计算机硬件系统之间的接口、OS 作为计算机系统资源的管理者、OS 用作扩充机器。

第二章

1.前趋图的定义：前趋图是一个有向无循环图 DAG。能更好的描述程序的顺序和并发执行情况

2.进程的定义：(1) 进程是程序的一次执行。(2) 进程是一个程序及其数据在处理器上顺序运行时所发生的活动。(3) 进程是程序在一个数据集合上的运行过程，它是系统进行资源分配和调度的一个独立单位。(4) 进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位

3.进程的特征：(1) 结构特征：程序段、数据段、PCB。(2) 动态性：进程是程序的一次运行过程，有生命周期。(3) 并发性。(4) 独立性：进程是一个能独立运行的基本单位，同时也是系统分配资源和调度的独立单位。(5) 异步性:进程按各自独立的、不可预知的速度向前推进。

4.进程三种基本状态：就绪状态；执行状态；阻塞状态。

5. 说明进程与程序的区别

(1) 程序是指令的有序集合，其本身没有任何运行的含义，它是一个静态的概念。而进程是程序在处理器上的一次执行过程，它是一个动态概念。

(2) 程序的存在是永久的。而进程则是有生命期的，它因创建而产生，因调度而执行，因得不到资源而暂停，因撤消而消亡。

(3) 程序仅是指令的有序集合。而进程则由程序段、相关数据段.进程控制块(PCB)组成。

(4) 进程与程序之间不是一一对应。

(5) 并发性：多个进程实体同存于内存中，且能在一段时间内同时运行。并发性是进程的重要特征，也是 OS 的重要特征。引入进程的目的也正是为了使其进程实体能和其它进程实体并发执行；而程序(没有建立 PCB)是不能并发执行的。

(6) 独立性：在传统的 OS 中，独立性是指进程实体是能独立分配资源和独立接受调度，能独立运行的基本单位。凡未建立 PCB 的程序都不能作为一个独立的单位参与运行。

(7) 异步性：进程实体按异步方式运行,推进次序每次不一定相同。

6.程序段、相关数据、PCB（进程控制块）三部分构成了进程实体。

7.进程控制块

概念：进程控制块是进程实体的重要组成部分，是操作系统中最重要的记录型数据，在进程控制块 PCB（Program Contral Block）中记录了操作系统所需要的、用于描述进程情况及控制进程运行所需要的全部信息

作用：通过 PCB，使得原来不能独立运行的程序（数据），成为一个可以独立运行的基本单位，一个能够并发执行的进程。进程控制块是进程存在的唯一标志。

8.引起挂起状态的原因有：终端用户的请求、父进程的请求、负荷调节的需要、操作系统的需要。

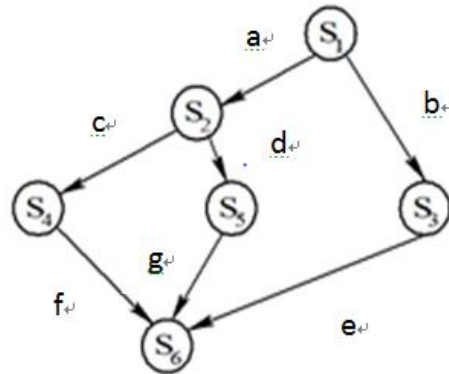
9.原语(Primitive)是由若干条指令组成的，用于完成一定功能的一个过程，原语在执行中不允许被中断，原语的作用是实现进程的通信和控制。常见的几种元语：创建原语 create()（功能：创建一新进程）、阻塞原语 block()（功能：将进程由执行状态转为阻塞状态）、唤醒原

语 wakeup() (功能: 将进程由阻塞状态变为就绪状态)、挂起原语 suspend() (功能: 将指定进程或处于阻塞状态的进程挂起)、激活原语 active() (功能: 将指定进程激活)。

6、同步机制应遵循的四条规则是: 空闲让进、忙则等待、有限等待、让权等待。

★7、试写出相应的程序来描述右图所示的前驱图

```
Var a,b,c,d,e,f,g: semaphore:=0,0,0,0,0,0,0;  
begin  
  parbegin  
    begin S1; signal(a); signal(b); end;  
    begin wait(a); S2; signal(c); signal(d); end;  
    begin wait(b); S3; signal(e); end;  
    begin wait(c); S4; signal(f); end;  
    begin wait(d); S5; signal(g); end;  
    begin wait(e); wait(f); wait(g); S6; end;  
  parend  
end
```



11.管程的定义: 一个管程定义了一个数据结构和能为并发进程所执行的一组操作, 这组操作能同步进程和改变管程中的数据。

12.线程的定义: 线程是进程中可独立执行的子任务, 是系统独立调度和分派的基本单位。

13.线程与进程的比较: a.拥有资源: 线程几乎不占资源, 同一进程的线程共享进程的资源

b.调度: 进程不再是调度的基本单位 c.并发性: 进程之间可以并发, 线程之间也可以并发执行

d.系统开销: 线程的创建、撤消与切换的系统开销小的多。

第三章

1.一个作业从提交开始, 往往要经历三级调度: 高级调度、低级调度、中级调度。

高级调度: 又称为作业调度或长程调度, 其主要功能是根据某种算法, 把外存上处于后备队列中的那些作业调入内存。它调度的对象是作业。

低级调度: 又称进程调度或短程调度。主要功能: 保存处理机的现场信息、按某种算法选取进程、把处理器分配给进程。常采用非抢占(非剥夺)方式和抢占(剥夺)方式两种。它所调度的对象是进程(或内核级线程)。

中级调度: 又称中程调度。在内存和外存对换区之间按照给定的原则和策略选择进程对换, 以解决内存紧张问题, 从而提高内存的利用率和系统吞吐量。

2.常见的调度算法有: 先来先服务调度算法(FCFS)、短作业/进程优先调度算法(SJF/SPF)、高优先权优先调度算法、基于时间片的轮转调度算法。

3. 优先权 = (等待时间 + 服务时间) / 服务时间。R 越大先执行

4.死锁的定义: 指多个进程在运行过程中因争夺资源而造成的一种僵局(deadly-Embrace), 若无外力作用, 这些进程都将无法向前推进。

5.产生死锁的原因有两个: 竞争资源、进程间推进顺序非法。

6. 产生死锁的四个必要条件是互斥条件、请求和保持条件、不剥夺条件、循环等待条件。

7.处理死锁的基本方法是: 预防死锁、避免死锁、检测死锁、解除死锁。

8.确定系统处于 S 为死锁状态的充分条件是: 当且仅当 S 状态的资源分配图是不可完全简化的。

9.常用的解除死锁方法有两种: 资源剥夺法、撤消进程法。

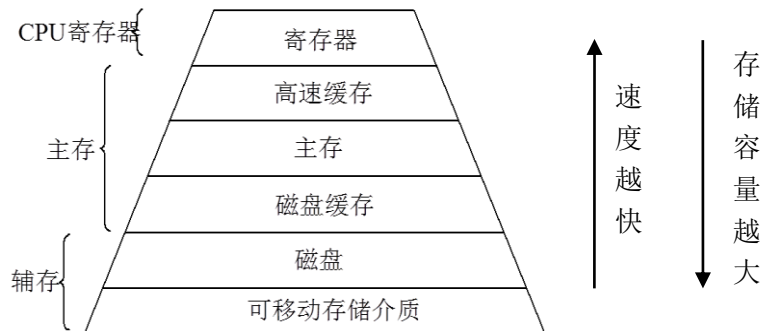
资源剥夺法: 当发现死锁后, 从其进程剥夺足够数量的资源给死锁进程, 以解除死锁状态。

撤消进程法: 采用强制手段从系统中撤消一个/一部分死锁进程, 并剥夺这些进程的资源

供其它死锁进程使用。

第四章

1、多级存储器结构



2. 存储器管理的目的：主存的分配和管理；提高主存储器的利用率；“扩充”主存容量；存储保护。

3. 动态存储分配时，要靠硬件地址变换机构实现重定位。

4. 为了实现进程对换，系统必须实现三方面功能：对换空间的管理、进程的换入和进程的换出。

5. 分页的作用是实现从页号到物理块号的地址映射。

6. 简述分页和分段有何相同点和不同点

相同点：分页和分段都采用**离散分配**的方式，且都要通过地址映射机构来实现地址变换

不同点：

(1) 从功能上看，**页是信息的物理单位**，分页是为实现离散分配方式，以消减内存的外零头，提高内存的利用率，即满足系统管理的需要，而不是用户的需要；而**段是信息的逻辑单位**，它含有一组其意义相对完整的信息，目的是为了能更好地满足用户的需要。

(2) 页的大小固定且由系统确定，而段的长度却不固定，决定于用户所编写的程序。

(3) 分页的作业地址空间是一维的，而分段的作业地址空间是二维的。

7. 页表（页面映象表）（1）页表的作用是实现从**页号到物理块号的地址映射**

（2）由页号和块号组成，指出逻辑地址中页号与主存中物理块号的对应关系

第五章

1. 虚拟存储器的定义：虚拟存储器是指仅把作业的一部分装入内存便可运行作业的存储管理系统，它具有请求页调入功能和页置换功能，能从逻辑上对内存容量进行扩充。

2. 虚拟存储器具有多样性、对换性、虚拟性三大主要特征。

3. 在存储管理中常用虚拟存储器方式来摆脱主存容量的限制。

4. 局部性原理：在一较短的时间内，程序的执行仅局限于某个部分，所访问的存储空间局限于某个区域。（2）时间局部性、空间局部性

5. 页表机制：

页号 物理块号 状态位 P 访问字段 A 修改位 M 外存地址

6. 改进型 clock 置换算法

下面四种类型的页面：

1 类 (A=0, M=0) 最近既未被访问，又未被修改；最佳淘汰页

2 类 (A=0, M=1) 最近既未被访问，但已被修改；并不是很好的淘汰页

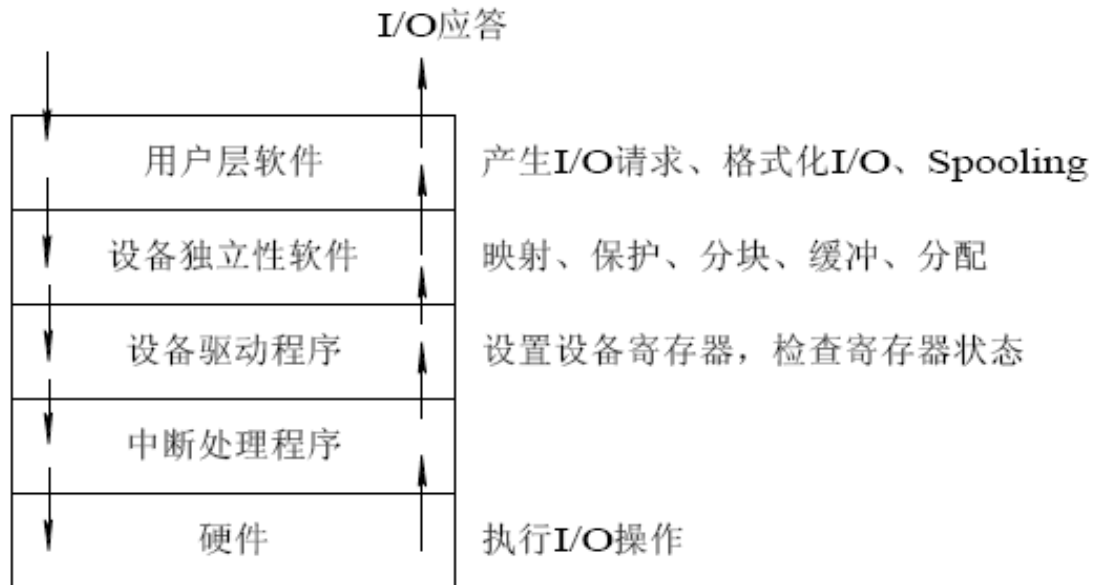
3 类 (A=1, M=0) 最近被访问，未被修改；该页有可能再被访问

4 类 (A=1, M=1) 最近被访问，且被修改。该页可能再被访问

第六章

1、I/O 系统的基本功能：隐藏物理设备的细节；与设备的无关性；提高处理机和 I/O 设备的利用率；对 I/O 设备进行控制；确保对设备的正确共享；错误处理。

2



I/O 系统的层次及功能

3、I/O 系统的分层：中断处理程序；设备驱动程序；设备独立性软件。

4、I/O 设备的类型

按传输速率分类：低速设备、中速设备、高速设备

按信息交换的单位分类：块设备、字符设备

按设备的共享属性分类：独占设备、共享设备、虚拟设备

5、I/O 通道定义：通道是独立于 CPU 的专门负责数据输入/输出传输工作的处理机，对外部设备实现统一管理，代替 CPU 对输入/输出操作进行控制，从而使输入，输出操作可与 CPU 并行操作。

6.外部设备引起。陷入：CPU 内部事件引起

7 所谓**设备控制器**，是一块能控制一台或多台外围设备与 CPU 并行工作的硬件

8.设备独立性：指用户设备独立于所使用的具体物理设备。即在用户程序中要执行 I/O 操作时，只需用逻辑设备名提出 I/O 请求，而不必局限于某特定的物理设备。

9、为什么要引入设备独立性？如何实现设备独立性？

引入设备独立性，可使应用程序独立于具体的物理设备，是设备分配具有灵活性。另外容易实现 I/O 重定向。

为了实现设备独立性，必须在设备驱动程序之上设置一层设备独立性软件，用来执行所有 I/O 设备的公用操作，并向用户层软件提供统一接口。关键是系统中必须设置一张逻辑设备表 LUT 用来进行逻辑设备到物理设备的映射，其中每个表目中包含了逻辑设备名、物理设备名和设备驱动程序入口地址三项；当应用程序用逻辑设备名请求分配 I/O 设备时，系统必须为它分配相应的物理设备，并在 LUT 中建立一个表目，以后进程利用该逻辑设备名请求 I/O 操作时，便可从 LUT 中得到物理设备名和驱动程序入口地址。

10、主存储器与外围设备之间的数据传送控制方式有程序直接控制、**中断驱动方式**、**DMA**

方式和通道控制方式。

11、虚拟设备是指采用 SPOOLing 技术，将某个独享设备改进为供多个用户使用的共享设备。

12、使用缓冲区能有效地缓和 I/O 设备和 CPU 之间速度不匹配的矛盾。

13、缓冲区的设置可分为单缓冲、双缓冲、多缓冲和缓冲池。

14、SPOOLing：即同时联机外围操作，又称脱机操作。在多道程序环境下，可利用多道程序中的一道程序，来模拟脱机的输入输出功能。即在联机条件下，将数据从输入设备传送到磁盘，或从磁盘传送到输出设备。

15、SPOOLing 系统特点：提高了 I/O 的速度、将独占设备改造成共享设备、实现了虚拟设备功能。

第七章

1、文件的概念：

定义：是指记录在外存上的具有文件名的一组相关信息的集合

文件属性：文件名、文件类型、文件长度、文件的物理位置、文件的建立日期以及用户对该文件的存取权限等

文件的特点：文件具有保存性、文件是按名存取、文件的内容是一组信息的集合

2、文件的逻辑结构分为流式文件（无结构文件）和记录式文件（有结构文件）二种。

3、文件的物理结构分为顺序文件、索引文件和索引顺序文件三种。

4、文件系统中，用于文件的描述和控制并与文件一一对应的是文件控制块。

5、对目录进行查询的方式常见的有两种：线性检索法和 Hash 方法。

6、目前广泛采用的目录结构是树型目录结构。

7、Hash 检索法有何优点？又有何局限性？

在 Hash 检索法中，系统利用用户提供的文件名并将它变换为文件目录的索引值，再利用该索引值到目录中去查找，这样能有效地提高目录的检索速度，但 Hash 检索法也有局限性即对于使用了通配符的文件名，系统是无法使用 Hash 检索法检索目录的。

名词解释

1.工作集：是指在某段时间间隔 Δ 里，进程实际所要访问页面的集合。

2.文件：文件是指由创建者所定义的、具有文件名的一组相关元素的集合，可分为有结构文件和无结构文件两种。在有结构文件中，文件由若干个相关记录组成，而无结构文件则被看成是一个字符流。文件在文件系统中是一个最大的数据单位，它描述了一个对象集。

3.位示图：位示图是利用二进制的一位来表示磁盘中的一个盘块的使用情况。当其值为“0”时，表示对应的盘块空闲；为“1”时，表示已经分配。有的系统把“0”作为盘块已分配的标记，把“1”作为空闲标志。（它们的本质上是相同的，都是用一位的两种状态标志空闲和已分配两种情况。）磁盘上的所有盘块都有一个二进制位与之对应，这样，由所有盘块所对应的位构成一个集合，称为位示图。

4.事务：事务是用于访问和修改各种数据项的一个程序单位。事务也可以被看做是一系列相关读和写操作。被访问的数据可以分散地存放在同一个文件的不同记录中，也可以放在多个文件中。只有对分布在不同位置的同一数据所进行的读和写（含修改）操作全部完成时，才能提交操作，结束事务，确认事务的变化。

5.检查点：当系统发生故障时，必须去检查整个 Log 表，以确定哪些事务需要利用 redo<Ti>过程去设置新值，而哪些事务又需要利用 undo<Ti>过程去恢复数据的旧值。由于在系统中可能存在着许多并发执行的事务，所以引入了检查点来对事务记录表中事务记录的清理工作经常化，保证系统的正常运行。

