

《计算机操作系统》学科复习强化版

◎第一章 操作系统引论

- ❖ **本章内容：**操作系统的目标和作用、操作系统的发展过程、操作系统的基本特征、操作系统的主要功能、OS 结构设计
 - ❖ **本章主要考点：**概念
 - 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。
 - 5、以下不是微内核 OS 特点的是 (C)
 - A、足够小的内核
 - B、应用“机制与策略分离”的原理
 - C、运行效率高
 - D、采用面向对象技术
- 注：微内核 OS 运行效率并不高，它还有一个特点是基于客户/服务器模式

★第二章 进程管理

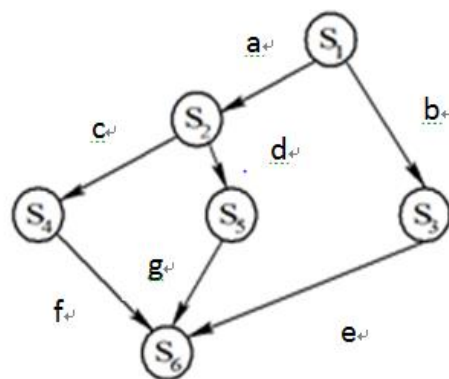
- ❖ **本章内容：**进程的基本概念、进程控制、进程同步、经典进程的同步问题、进程通信、线程的基本概念
- ❖ **主要考点：**★程序和进程的区别、进程的状态转化、原语、★进程同步 S 值的变化和前趋图
管程、线程的概念
- 1、简述进程的定义
 - (1) 进程是程序的一次执行。
 - (2) 进程是一个程序及其数据在处理机上顺序执行时所发生的活动。
 - (3) 进程是程序在一个数据集合上的运行过程，它是系统进行资源分配和调度的一个独立单位。
 - (4) 进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位
- ★2、说明进程与程序的区别
 - (1) 程序是指令的有序集合，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机上的一次执行过程，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。
- 3、程序段、相关数据、PCB（进程控制块）三部分构成了进程实体。
- 4、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。
- 5、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
```



8、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。

9、高级通信机制分为三大类：共享存储器系统、消息传递系统、管道通信系统

10、操作系统中引入进程的目的是为了使多个程序能并发执行，以提高资源利用率和系统吞吐量，在操作系统中再引入线程，则是为了减少程序在并发执行时所付出的时空开销。

11、进程与线程的比较

	进程	线程
引入目的	能并发执行,提高资源的利用率和系统吞吐量.	提高并发执行的程度,减小开销,进一步提高资源的利用率和系统吞吐量.
并发性	较低	较高
基本属性(调度)	资源拥有的基本单位—进程	独立调度/分派的基本单位—线程
基本状态	就绪;执行;等待	就绪;执行;等待
拥有资源	资源拥有的基本单位—进程	无资源分配
系统开销	创建/撤消/ 切换 时空开销较大	创建/撤消/切换时空开销较小
系统操作	创建,撤消,切换	创建,撤消,切换
存在标志	进程控制块PCB	线程控制块TCB
关系	单进程单线程;单进程多线程;多进程单线程;多进程多线程	

12、试从调度性，并发性，拥有资源及系统开销方面对进程和线程进行比较。

调度性：在传统的操作系统中，拥有资源的基本单位和独立调度、分派的基本单位都是进程，在引入线程的 OS 中，则把线程作为调度和分派的基本单位，而把进程作为资源拥有的基本单位；

并发性：在引入线程的 OS 中，不仅进程之间可以并发执行，而且在一个进程中的多个线程之间，亦可并发执行，因而使 OS 具有更好的并发性；

拥有资源：无论是传统的操作系统，还是引入了线程的操作系统，进程始终是拥有资源的一个基本单位，而线程除了拥有一点在运行时必不可少的资源外，本身基本不拥有系统资源，但它可以访问其隶属进程的资源；

开销：由于创建或撤销进程时，系统都要为之分配和回收资源，如内存空间等，进程切换时所保存和设置的现场信息也要明显地多于线程，因此，操作系统在创建、撤销和切换进程时所付出的开销将显著地大于线程。

13、当一个进程完成了特定任务后，系统回收这个进程所占的主存空间和取消该进程的进程控制块 (PCB)就撤销了该进程。

14、当一个进程独占处理器顺序执行时，具有两个特性：封闭性和可再现性。

15、对信号量 S 的操作只能通过原语操作进行，对应每一个信号量设置了一个等待队列。

16、在操作系统中，进程是一个资源分配的基本单位，也是一个独立运行和调度的基本单位。

★第三章 处理机调度与死锁

❖ **本章内容：**处理机调度的层次、调度准则、调度算法、产生死锁的原因和必要条件、预防死锁的方法、死锁的检测与解除

❖ **主要考点：**处理机调度的三个层次、调度准则、调度算法、实时调度算法、产生死锁的原因和必要条件、★★**利用银行家算法避免死锁**、死锁的检测与解除

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、选择调度方式和调度算法的准则

面向用户的准则：(1) 周转时间短(2) 响应时间快(3) 截止时间的保证(4) 优先权准则

面向系统的准则：(1) 系统吞吐量高(2) 处理机利用率好(3) 各类资源的平衡利用

3、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。

4、进程调度算法采用时间片轮转法时，时间片过大会使轮转法转为先来先服务调度算法 (FCFS)。

5、若使当前运行的进程总是优先级最高的进程，则应该选择进程高优先权优先调度算法

6、在响应比最高者优先的作业调度算法中，当各个作业等待时间相同时，运行时间短的作业将得到优先调度；当各个作业要求运行的时间相同时，等待时间长的作业得到优先调度。

7、一个理想的作业调度算法应该是既能提高系统的效率，又能使系统的作业及时得到结果。

8、常用的几种实时调度算法有：最早截止时间优先算法 (EDF)、最低松弛度优先算法 (LLF)

最早截止时间优先算法：该算法是根据任务的开始截止时间来确定任务的优先级。开始截止时间越早，其优先级越高

最低松弛度优先算法：该算法是根据任务紧急（或松弛）的程序，来确定任务的优先级。任务的紧急度越高，其优先级越高，并使之优先执行

9、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。

10、产生死锁的原因有两个：竞争资源、进程间推进顺序非法。

★11、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。

12、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。

13、什么是系统的安全状态，避免死锁的实质是什么？

系统的安全状态是指在某一时刻,系统能按某种进程顺序(p_1, p_2, \dots, p_n)来为每个进程 P_i 分配其资源,直到满足每个进程对资源的最大需求,使每个进程都可顺利地完 成,则称此时的系统状态为安全状态.称序列 $\langle p_1, p_2, \dots, p_n \rangle$ 为安全序列。如果一个系统在安全状态,就没有死锁。如果一个系统处于不安全状态,就有可能死锁。**避免死锁的实质: 确保系统不进入不安全状态。**

★★14、利用银行家算法避免死锁

●储备知识

银行家算法中的数据结构：假定系统中有 n 个进程 (P_1, P_2, \dots, P_n)， m 类资源 (R_1, R_2, \dots, R_m)，银行家算法中使用的数据结构如下：

▲可利用资源向量: $Available[j]=K$, 表示系统资源 R_j 类资源有 K 个可用

▲最大需求矩阵: $\text{Max}[i,j]=K$, 表示进程 P_j 需要 R_i 类资源的最大数目是 K 个

▲分配矩阵: $\text{Allocation}[i,j]=K$, 表示进程 P_i 当前已经分配到 K 个 R_j 类资源

▲需求矩阵: $\text{Need}[i,j]=K$, 表示进程 P_i 还需要 K 个 R_j 类资源方能完成任务

三个矩阵的关系:

Need $[i,j] = \text{Max}[i,j] - \text{Allocation}[i,j]$.

银行家算法：设 $Request_i$ 是进程 P_i 的请求向量，设 $Request_i[j] = K$ ，表示进程 P_i 请求分配 R_j 类资源 K 个。当进程 P_i 发出资源请求后，系统按如下步骤进行检查：

(1) 如 $Request[i] \leq Need[i, j]$, 转(2); 否则出错, 因为进程申请资源量超过它声明的最大量。

(2) 如 $Request[i] \leq Available[j]$, 转(3); 否则表资源不够, 需等待。

(3)系统**试分配**资源给进程 P_i ,并作如下修改:

$$\text{Available}[j] = \text{Available}[j] - \text{Request}_i[j]$$

$$\text{Allocation}[i,j] = \text{Allocation}[i,j] + \text{Request}_i[j]$$

$$\text{Need}[i,j] = \text{Need}[i,j] - \text{Request}_i[j]$$

(4)系统执行安全性算法,检查此次资源分配后,系统是否处于安全状态。若安全,则**正式**进行分配,否则恢复原状态让进程 P_i 等待。

安全性算法：为了进行安全性检查，需要定义如下数据结构：

(1) 工作向量 **Work**，它表示系统可以提供给进程继续运行所需的各类资源数目，它含有 m 个元素，在执行安全算法开始时， $\text{Work} \equiv \text{Available}$

(2) Finish, 它表示系统是否有足够的资源分配给进程, 使之运行完成。开始时, Finish[i]=false; 当有足够资源分配给进程 P_i 时, 令 Finish[i]=true。

安全性检查算法执行步骤:

(1) Work := Available

Finish [i] = false

(2)寻找满足如下条件的进程 P_i

Finish [i] = false

Need[i,j] ≤ Work[j], 如果找到, 转(3), 否则转(4)

(3)当进程 P_i 获得资源后,可顺利执行完,并释放分配给它的资源,故执行:

Work [j]= Work [j]+ Allocation[i,j];

```
Finish[i] = ture;
```

转 (2).

(4)若所有进程的 $Finish[i] = true$ ，则表示系统处于安全状态，否则处于不安全状态

●银行家算法例题：在银行家算法中，若出现下述资源分配情：

Process	Allocation	Need	Available
P ₀	0032	0012	1622
P ₁	1000	1750	
P ₂	1354	2356	
P ₃	0332	0652	
P ₄	0014	0656	

试问：

(1) 该状态是否安全？

(2) 若进程 P₂ 提出请求 Request(1, 2, 2, 2)后，系统能否将资源分配给它？

解：(1)该状态是安全的，因为存在一个安全序列< P₀P₃P₄P₁P₂>。下表为该时刻的**安全序列表**。

资源情况 进程	Work	Need	Allocation	Work+Allocation	Finish
P ₀	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	true
P ₃	1 6 5 4	0 6 5 2	0 3 3 2	1 9 8 6	true
P ₄	1 9 8 6	0 6 5 6	0 0 1 4	1 9 9 10	true
P ₁	1 9 9 10	1 7 5 0	1 0 0 0	2 9 9 10	true
P ₂	2 9 9 10	2 3 5 6	1 3 5 4	3 12 14 14	true

(2)若进程 P₂ 提出请求 Request(1, 2, 2, 2)后，系统不能将资源分配给它，若分配给进程 P₂，系统还剩下的资源情况为 (0, 4, 0, 0)，此时系统中的资源将无法满足任何一个进程的资源请求，从而导致系统进入不安全状态，容易引起死锁的发生。

(书上例题雷同，在 P110，可同步参考)

15、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。**

16、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。

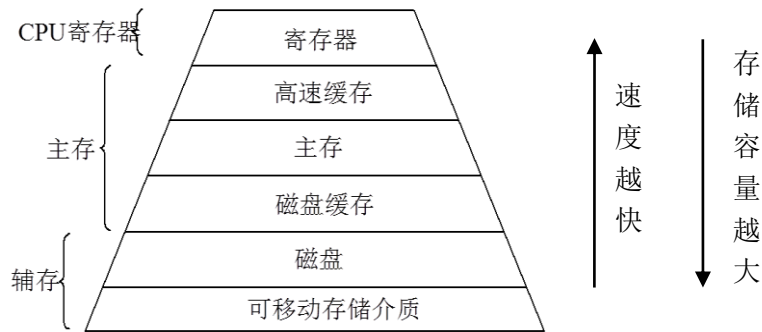
撤消进程法：采用强制手段从系统中撤消一个/一部分死锁进程,并剥夺这些进程的资源供其它死锁进程使用。

★第四章 存储器管理

❖ **本章内容：**存储器的层次结构、连续分配方式、基本分页存储管理方式、基本分段存储管理方式、虚拟存储器的基本概念、页面置换算法

❖ **主要考点：**存储器的层次结构、★地址变换、虚拟存储器的概念、特征、★★页面置换算法

1、多级存储器结构



2、在动态分区式内存分配算法中，倾向于优先使用**低地址部分**空闲区的算法是首次适应算法；能使内存空间中空闲区分布较均匀的算法是循环首次适应算法。

3、连续分配方式

固定分配方式	<p>存储管理方法:将内存用户空间分为若干固定大小的区域。除 OS 占一区外，其余的一个分区装入一道程序。分区的大小事先必须确定</p> <p>主要特点:管理简单，但因作业的大小并不一定与某个分区大小相等，从而使一部分存储空间被浪费。所以主存的利用率不高。</p>
	<p>存储管理方法:在作业进入内存时，根据作业的大小动态地建立分区，并使分区的大小正好适应作业的需要</p> <p>主要特点:管理简单，只需小量的软件和硬件支持，便于用户了解和使用。进程的大小与某个分区大小相等，从而主存的利用率有所提高。</p> <p>分区分配算法:首次适应算法、循环首次适应算法、最佳适应算法、最坏适应算法、快速适应算法</p>

4、动态存储分配时，要靠硬件地址变换机构实现**重定位**。

5、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。

6、分页的作用是实现从页号到物理块号的地址映射。

7、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。

8、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。

9、虚拟存储器具有**多样性、对换性、虚拟性**三大主要特征。

10、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。

★11、若在一分页存储管理系统中，假定某时刻该系统为用户的第 0, 1, 2, 3 页分别分配的物理地址为 2, 3, 1, 6。已知页面大小为 1KB，试将逻辑地址 1011 转化为相应的物理地址？

解：由题意逻辑地址：页号 p 为 2 位，位移量 w 为 10 位；（ $1KB=2^{10}B$ ）

物理地址：块号 b 为 3 位，块内位移 d 为 10 位

逻辑地址 1011（十进制）的二进制表示为 00 1111110011

由此可知逻辑地址 1011 的页号 0，该页放在第 2 物理块中，其物理地址的二进制表示为

010 1111110011

提示：对于逻辑地址计算：其中页内地址表每页的大小即 $1KB=2^{10}B$ ，所以页内地址（位移量 w ）为 10 位。其中页号表最多允许的页数即 4 页= 2^2 页，所以页号（ P ）为 2 位。对于物理地址计算：其中块内地址表每块的大小与页大小相等，所以块内地址（块内位移 d ）也为 10 位。其中块号表内存空间最大块数为 6，所以块号为 3 位（ $6 \leq 2^3=8$ ）

★★12、页面置换算法

常见的几种页面置换算法：最佳（optimal）置换算法、★先进先出页面置换算法（FIFO）、

★最近最久未使用置换算法（LRU）、Clock 置换算法

①最佳（optimal）置换算法

算法思想：选择的被淘汰页面，将是以后永不使用的，或许是在最长(未来)时间内不再被访问的页面。采用最佳置换算法，通常可保证获得最低的缺页率。但在现在中无法实现，但可以用该算法去评价其他算法。

例：假定系统为某进程分配了三个物理块，并考虑有以下的页面号引用串：7，0，1，2，0，3，0，4，2，3，0，3，2，1，2，0，1，7，0，1

页面请求	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
最佳置换	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
算法执行		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
过程			1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
注释	调	调	调	替	命	替	命	替	命	命	替	命	命	替	命	命	命	替	命	命
	入	入	入	换	中	换	中	换	中	中	换	中	中	换	中	中	中	换	中	中

由图看出，采用最佳置换算法发生了 6 次缺页中断，命中率为 55%

②先进先出（FIFO）页面置换算法

算法思想：该算法总是淘汰最先进入内存的页面，即选择在内存中驻留时间最久的页面予以淘汰。该算法实现简单，但与进程实际运行的规律不相适应，因为在进程中，有些页面经常被访问，比如，含有全局变量、常用函数、例程等的页面，FIFO 算法并不能保证这些页面不被淘汰。

例：假定系统为某进程分配了三个物理块，并考虑有以下的页面号引用串：7，0，1，2，0，3，0，4，2，3，0，3，2，1，2，0，1，7，0，1

页面请求	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
先进先出	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
置换算法		0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
执行过程			1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1
注释	调	调	调	替	命	替	替	替	替	替	替	命	命	替	替	命	命	替	替	替
	入	入	入	换	中	换	换	换	换	换	换	中	中	换	换	中	中	换	换	换

由图看出，采用最佳置换算法发生了 12 次缺页中断，命中率为 25%

③最近最久未使用置换算法（LRU）

算法思想：该算法根据页面调入内存后的使用情况进行决策的，当须淘汰一个页面时，选择最近最久未使用的页面予以淘汰。

例：假定系统为某进程分配了三个物理块，并考虑有以下的页面号引用串：7，0，1，2，0，3，0，4，2，3，0，3，2，1，2，0，1，7，0，1

页面请求	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
先进先出	7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
置换算法		0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
执行过程			1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
注释	调入	调入	调入	替换	命中	替换	命中	替换	替换	替换	替换	命中	命中	替换	命中	替换	命中	替换	命中	命中

由图看出，采用最佳置换算法发生了 9 次缺页中断，命中率为 40%

▲例：设某作业占有 7 个页面，如果在主存中只允许装入 4 个工作页面，作业运行时，实际访问的页面顺序为 1, 2, 3, 6, 4, 7, 3, 2, 1, 4, 7, 5, 6, 5, 2, 1。试用 FIFO 和 LRU 页面调度算法，列出各自的页面淘汰顺序和缺页中断次数，以及最后留驻主存 4 页的顺序（假设开始的 4 个页面已经装入主存）

页面请求	1	2	3	6	4	7	3	2	1	4	7	5	6	5	2	1
FIFO 置换算法执行过程				1	4	4	4	4	4	4	4	5	5	5	5	5
				2	2	7	7	7	7	7	7	7	6	6	6	6
				3	3	3	3	2	2	2	2	2	2	2	2	2
				6	6	6	6	6	1	1	1	1	1	1	1	1
注释				开始	替换	替换	命中	替换	替换	命中	命中	替换	替换	命中	命中	命中
LRU 置换算法执行过程				1	4	4	4	4	1	1	1	1	6	6	6	6
				2	2	7	7	7	7	4	4	4	4	4	2	2
				3	3	3	3	3	3	3	7	7	7	7	7	1
				6	6	6	6	2	2	2	2	5	5	5	5	5
注释				开始	替换	替换	命中	替换	替换	替换	替换	替换	替换	命中	替换	替换

FIFO 淘汰顺序：1, 2, 3, 6, 4, 7 缺页中断次数（替换次数）：6 次

最后留驻主存 4 页的顺序：5, 6, 2, 1

LRU 淘汰顺序：1, 2, 6, 4, 7, 3, 2, 1, 4, 7 缺页中断次数（替换次数）：10 次

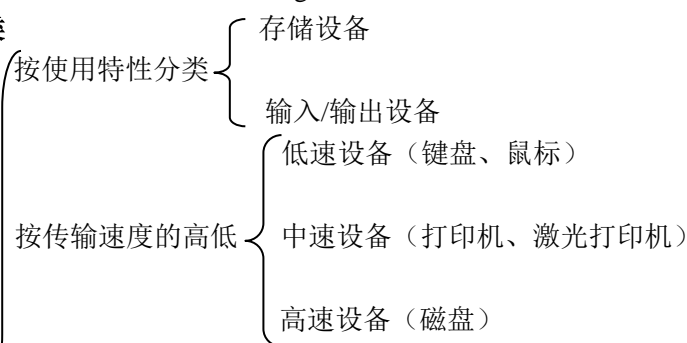
最后留驻主存 4 页的顺序：6, 2, 1, 5

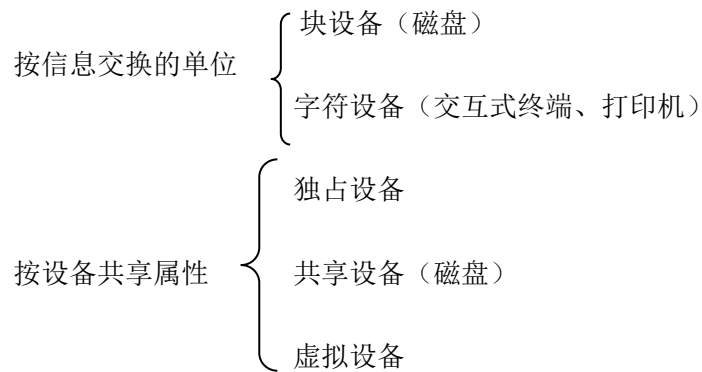
◎第五章 设备管理

❖ 本章内容：I/O 系统、I/O 控制方式、缓冲管理、设备分配、设备处理、磁盘存储器管理

❖ 主要考点：I/O 系统设备分类、SPOOLing 技术、基本概念

1、I/O 系统设备分类



I/O 系统设备分类

2、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

3、什么是设备独立性

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。

4、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。

5、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。

6、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。

7、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。

8、缓冲区的设置可分为**单缓冲**、**双缓冲**、**多缓冲**和**缓冲池**。

9、什么是 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。

10、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。**

◎第六章 文件管理

❖ **本章内容：**文件和文件系统、文件的逻辑结构、目录管理、

❖ **主要考点：**文件的概念、逻辑物理结构、目录管理、基本概念

1、文件的基本概念

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。

附：各章判断题汇总

- (×) 1、并发性是指若干事件在同一时刻发生。
- (√) 2、虚存容量的扩大是以牺牲 CPU 工作时间以及内、外存交换时间为代价的。
- (×) 3、用户为每个自己的进程创建 PCB，并控制进程的执行过程。
- (√) 4、树型目录结构能够解决文件重名问题。
- (√) 5、原语是一种不可分割的操作。
- (√) 6、通道一旦被启动就能独立于 CPU 运行，这样可使 CPU 和通道并行操作。
- (√) 7、页式的地址是一维的，段式的地址是二维的
- (×) 8、位示图方法可用于磁盘的调度管理。
- (×) 9、虚拟设备是指把一个物理设备变换成多个对应的逻辑设备，它通过逻辑设备表来实现的。
- (×) 10、页式管理易于实现不同进程间的信息共享。
- (√) 11、在虚拟存储方式下，程序员编制程序时不必考虑主存的容量，但系统的吞吐量在很大程度上依赖于主存储器的容量；
- (×) 12、可重定位分区管理可以对作业分配不连续的内存单元；
- (√) 13、采用动态重定位技术的系统，目标程序可以不经任何改动，而装入物理内存；
- (×) 14、页式存储管理中，一个作业可以占用不连续的内存空间，而段式存储管理，一个作业则是占用连续的内存空间。
- (×) 15、线程是最小的拥有资源的单位。
- (√) 16、文件系统最基本的功能是实现按名存取。
- (×) 17、存取控制表是每个用户一张，表明该用户对不同文件的存取权限。
- (×) 18、SPOOLing 技术可以解决进程使用设备死锁问题。
- (×) 19、对于一个具有三级索引表的文件，存取一个记录需要访问三次磁盘。
- (√) 20、在 I/O 控制的多种方式中，传输速率高，对主机影响少的方式最好。
- (×) 21、进程可以删除自己的 PCB 表。
- (×) 22、可重定位分区法能够支持虚拟存储器的技术。
- (×) 23、单级目录结构能够解决文件重名问题。
- (×) 24、分页式存储管理中，页的大小是可以不相等的。
- (√) 25、执行原语时不会响应任何中断。
- (√) 26、段页式管理实现了段式、页式两种存储方式的优势互补。
- (√) 27、对临界资源应采取互斥访问方式来实现共享。
- (×) 28、文件系统中分配存储空间的基本单位是记录。
- (×) 29、外存对换空间保存的是虚拟内存管理系统调出的程序。
- (√) 30、虚存容量的扩大是以牺牲 CPU 工作时间以及内、外存交换时间为代价的