

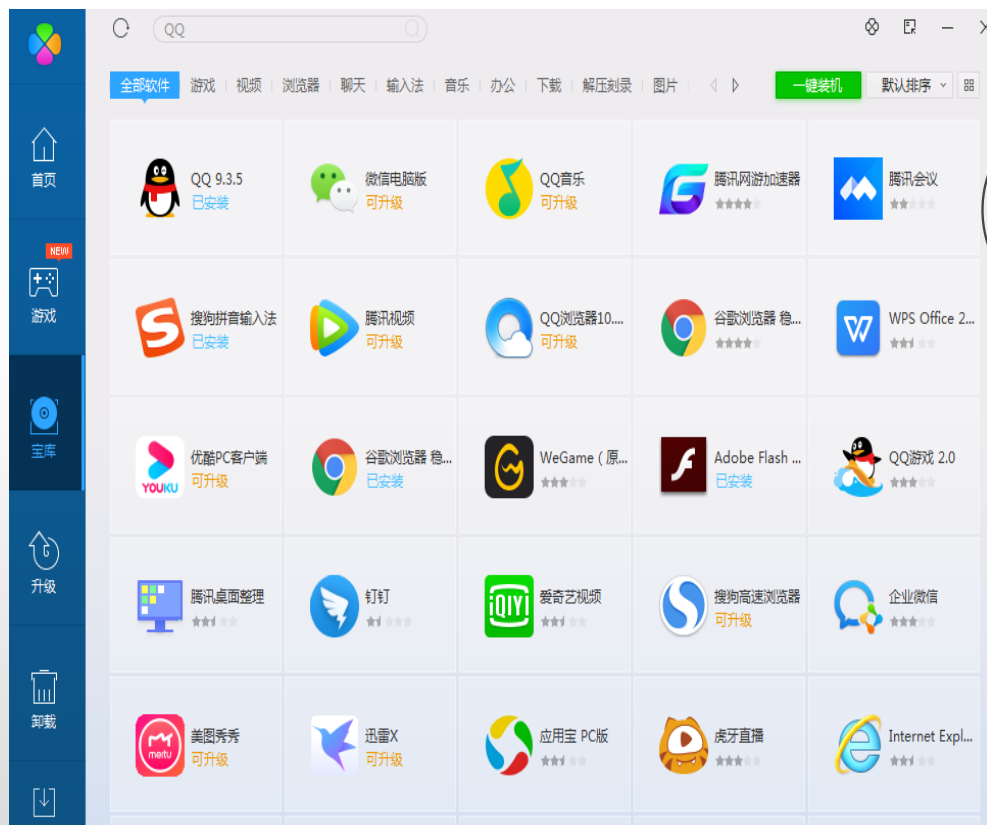
操作系统基础

51CTO学院：邹月平

操作系统概述



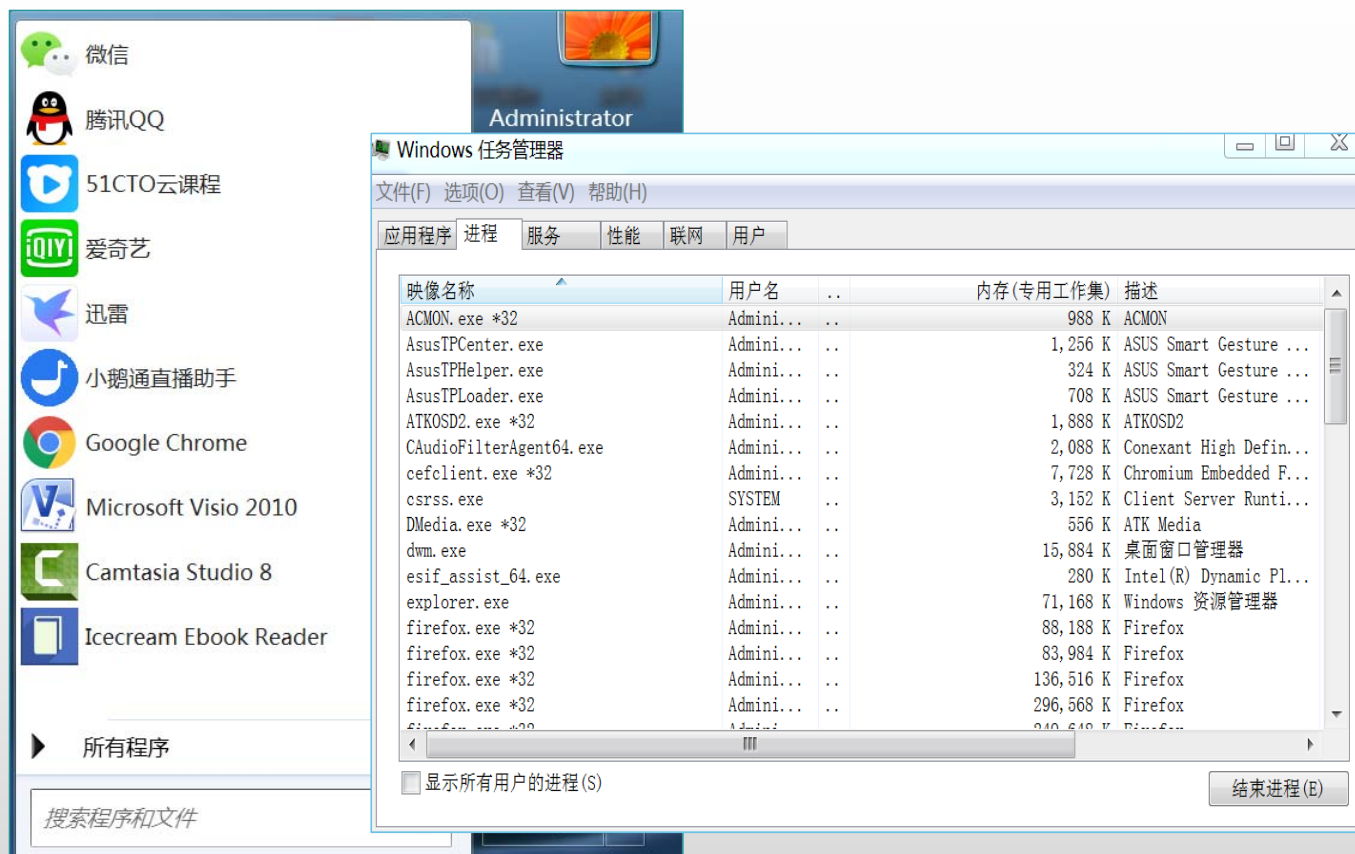
操作系统概述



听歌、上网、
玩游戏...通
通不行...



操作系统概述

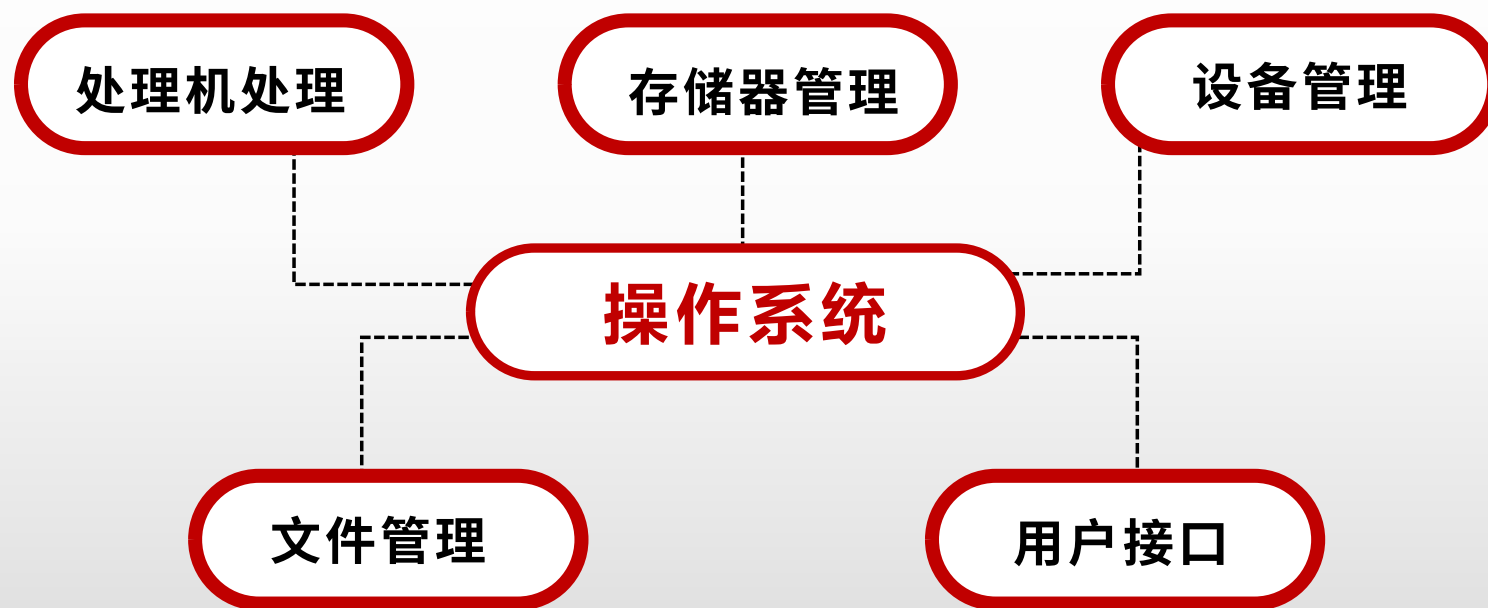


操作系统概述

```
1  #include <stdio.h>
2  //Hello.c→Hello.exe
3  int main(void) {
4      char *Hello="Hello"
5      printf("%s",Hello);
6      while(TRUE)
7      {
8          //死循环
9          int i=100;
10     }
11 }
12
```

- Hello.exe文件的存放
- Hello.exe程序的启动
- 为Hello.exe分配内存
- printf输出字符串
- 死循环对CPU的占用
- 程序结束退出系统

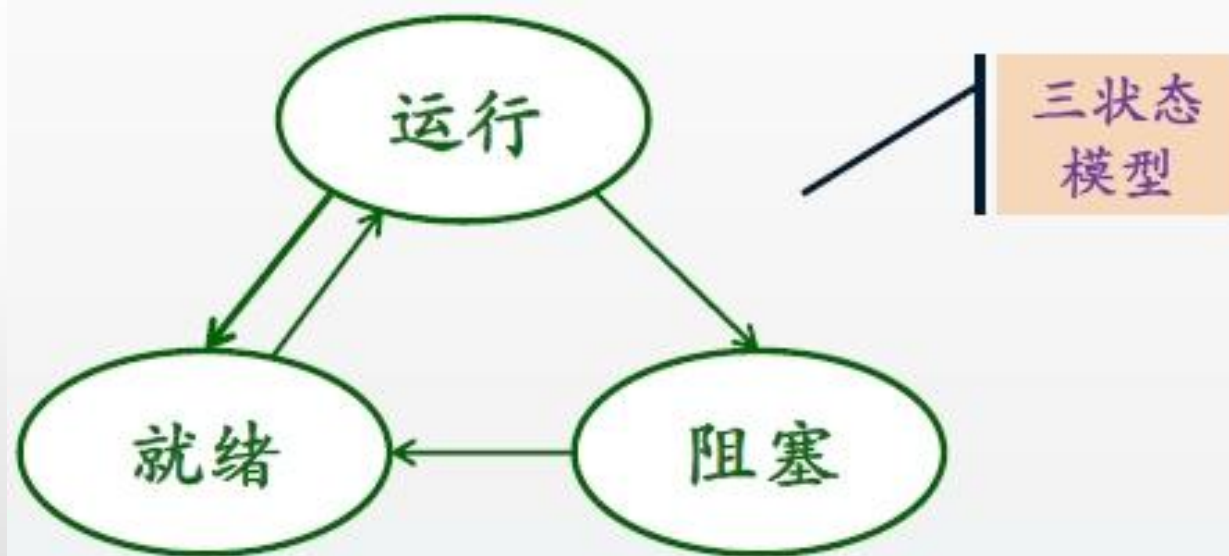
操作系统概述



考点分布

- ▶ **处理机处理** 进程的状态、前驱图、PV操作
- ▶ **存储器管理** 逻辑地址、物理地址、存储方案
- ▶ **设备管理** 输入输出控制方式
- ▶ **文件管理** 文件的索引
- ▶ **用户接口**

进程

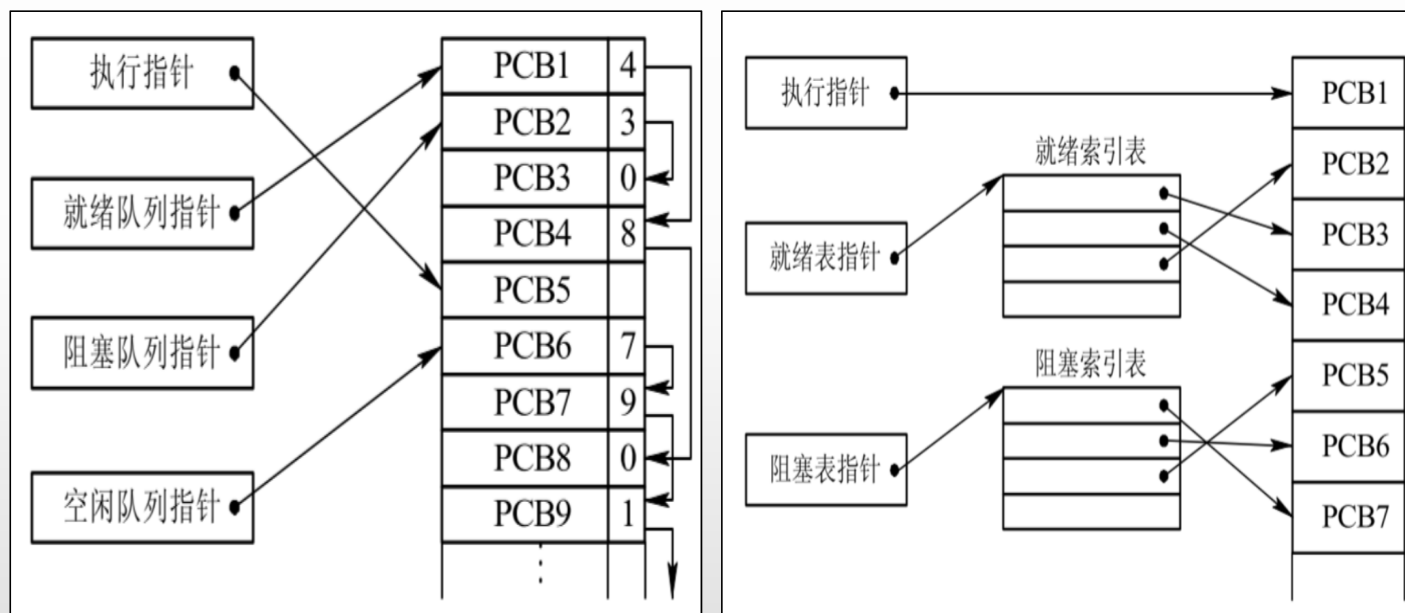


进程

进程通常由程序、数据集合、进程控制块 **PCB** 组成。**PCB** 是一种数据结构，是进程存在的唯一标识。

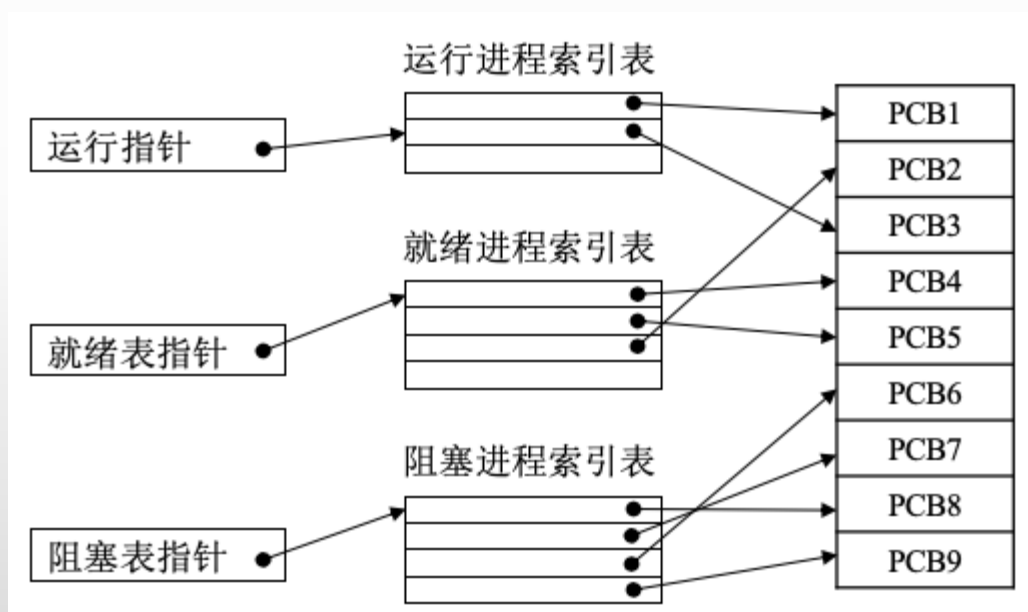
线性方式	把所有PCB组织在一张线性表中，每次查找是需要扫描全表。
链接方式	把具有同一状态的 PCB，用其中的链接字链接成一个队列，PCB存储在一个连续的区域。
索引方式	同一状态的进程归入一个索引表，多个状态对应多个不同的索引表。

进程



典型真题

某计算机系统中的进程管理采用三态模型，那么下图所示的PCB（进程控制块）的组织方式采用（ ），图中（ ）。



典型真题

A. 顺序方式 B. 链接方式 C. 索引方式 D. Hash

- A. 有1个运行进程，2个就绪进程，4个阻塞进程
- B. 有2个运行进程，3个就绪进程，3个阻塞进程
- C. 有2个运行进程，3个就绪进程，4个阻塞进程
- D. 有3个运行进程，2个就绪进程，4个阻塞进程

典型真题

试题分析

进程控制块PCB的组织方式有：线性表方式，索引表方式，链接表方式。

线性表方式：不论进程的状态如何，将所有的PCB连续地存放在内存的系统区。这种方式适用于系统中进程数目不多的情况。

索引表方式：该方式是线性表方式的改进，系统按照进程的状态分别建立就绪索引表、阻塞索引表等。

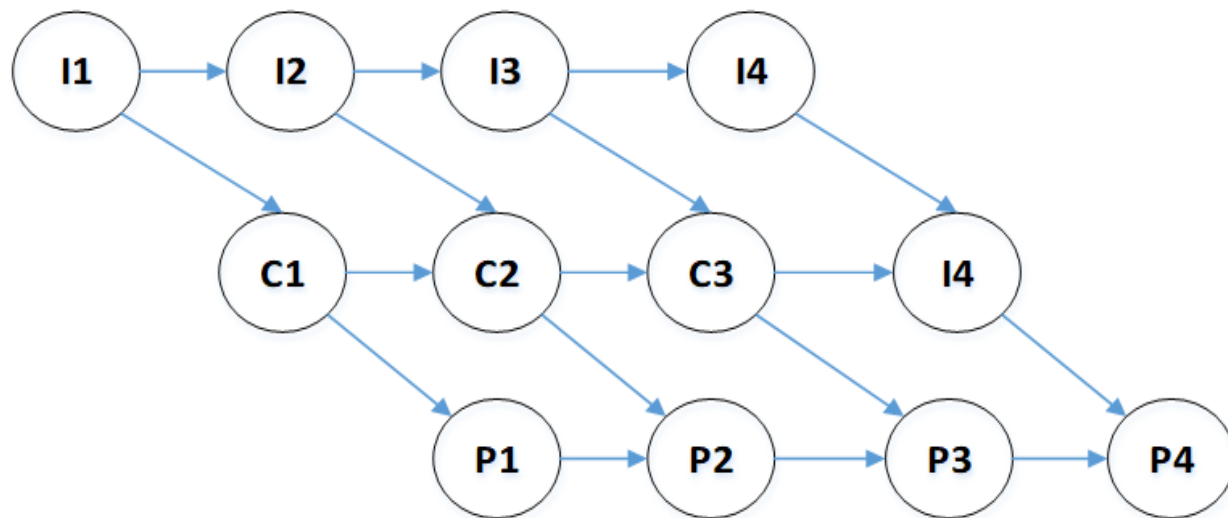
链接表方式：系统按照进程的状态将进程的PCB组成队列，从而形成就绪队列、阻塞队列、运行队列等。

运行进程：PCB1、PCB3，就绪进程：PCB2、PCB4、PCB5，阻塞进程：PCB6、PCB7、PCB8、PCB9。

参考答案：C、C

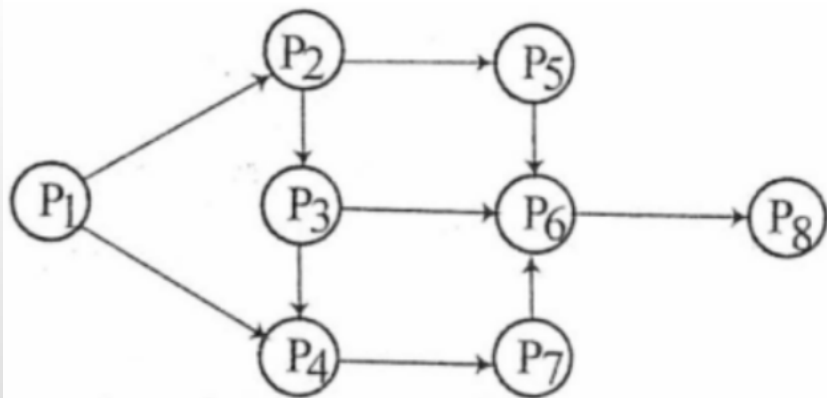
进程

前驱图是一个有向无循环图，由节点和有向边组成，节点代表各程序段的操作，而节点间的有向边表示两个程序段操作之间存在的前趋关系。用于这种图可以描述多个程序或进程之间的执行顺序关系。



典型真题

前驱图是一个有向无环图,记为 $\rightarrow = \{p_i, p_j, p_i \text{完成时间先于} p_j \text{开始时间}\}$ 。假设系统中进 $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$, 且进程的前驱图如下。那么该前驱图可记为 (1) 图中 (2) 。



典型真题

(1) $A \rightarrow = \{(P1, P2), (P1, P3), (P1, P4), (P2, P5), (P3, P2), (P3, P4), (P3, P6), (P4, P7), (P5, P8)\}$

$B \rightarrow = \{(P1, P2), (P1, P4), (P2, P3), (P2, P5), (P3, P4), (P3, P6), (P4, P7), (P5, P6), (P7, P6), (P6, P8)\}$

$C \rightarrow = \{(P1, P2), (P1, P4), (P2, P5), (P3, P2), (P3, P4), (P3, P6), (P4, P6), (P4, P7), (P6, P8), (P7, P8)\}$

$D \rightarrow = \{(P1, P2), (P1, P3), (P2, P4), (P2, P5), (P3, P2), (P3, P4), (P3, P5), (P4, P7), (P6, P8), (P7, P8)\}$

(2) A.存在着十个前驱关系。p1为初始节点。p2p4为终止节点。

B.存在着两个前驱关系。p6为初始节点，p2p4为终止节点。

C.存在着九个前驱关系。p6为初始节点，p8为终止节点。

D.存在着十个前驱关系。p1为初始节点，p8为终止节点。

典型真题

试题分析

本题考查前趋图相关知识。

前趋图标记规则：从小到大。使用排除法，解题思路如下：

- 1) A选项, (P5,P8)不存在, 排除
- 2) C选项, (P2,P3)缺失, 排除
- 3) D选项, (P2,P4)不存在, 排除
- 4) 只剩下B选项, 经检验, 正确。

观察B选项, 共有10对前驱关系, P1为初始节点, P8为终止节点。

参考答案： (1) B、 (2) D

PV操作

i为全局变量

程序A

- 1)
- 2) **i=100;**
- 3)
- 4) **printf("A:i=%d,i)**
- 5)
- 6)

程序B

- 1)
- 2) **i=200;**
- 3)
- 4) **printf(" A:i=%d,i)**
- 5)
- 6)

执行结果1	A: i=100	√	B: i=200	√
执行结果2	A: i=200	×	B: i=200	√
执行结果3	A: i=100	√	B: i=100	×

PV操作

□ 互斥问题:

- 进入临界区之前先执行P操作（可能阻塞当前进程）
- 离开临界区之后执行V操作；（可能唤醒某个进程）

```
main( )
{
    /*信号量mutex*/
    int mutex=1;
    cobegin//并发
    Pa( );
    Pb( );
    Pc( );
    coend//并发结束
}
```

```
Pb( )
{
    P(mutex);
    CSb
    V(mutex);
}
```

```
Pa( )
{
    P(mutex);
    CSa
    V(mutex);
}
```

```
Pa( )
{
    P(mutex);
    CSc
    V(mutex);
}
```

PV操作

P操作：

- ①将信号量S的值减1，即 $S=S-1$ ；
- ②如果 $S \geq 0$ ，则该进程继续执行；否则该进程置为等待状态。

V操作：

- ①将信号量S的值加1，即 $S=S+1$ ；
- ②如果 $S > 0$ 该进程继续执行；否则说明有等待队列中有等待进程，需要唤醒等待进程。

PV操作

int mutex=1;

```
Pa( )  
{  
    P(mutex);  
    CSa  
    V(mutex);  
}
```

mutex:1→0

```
Pb( )  
{  
    P(mutex);  
    CSb  
    V(mutex);  
}
```

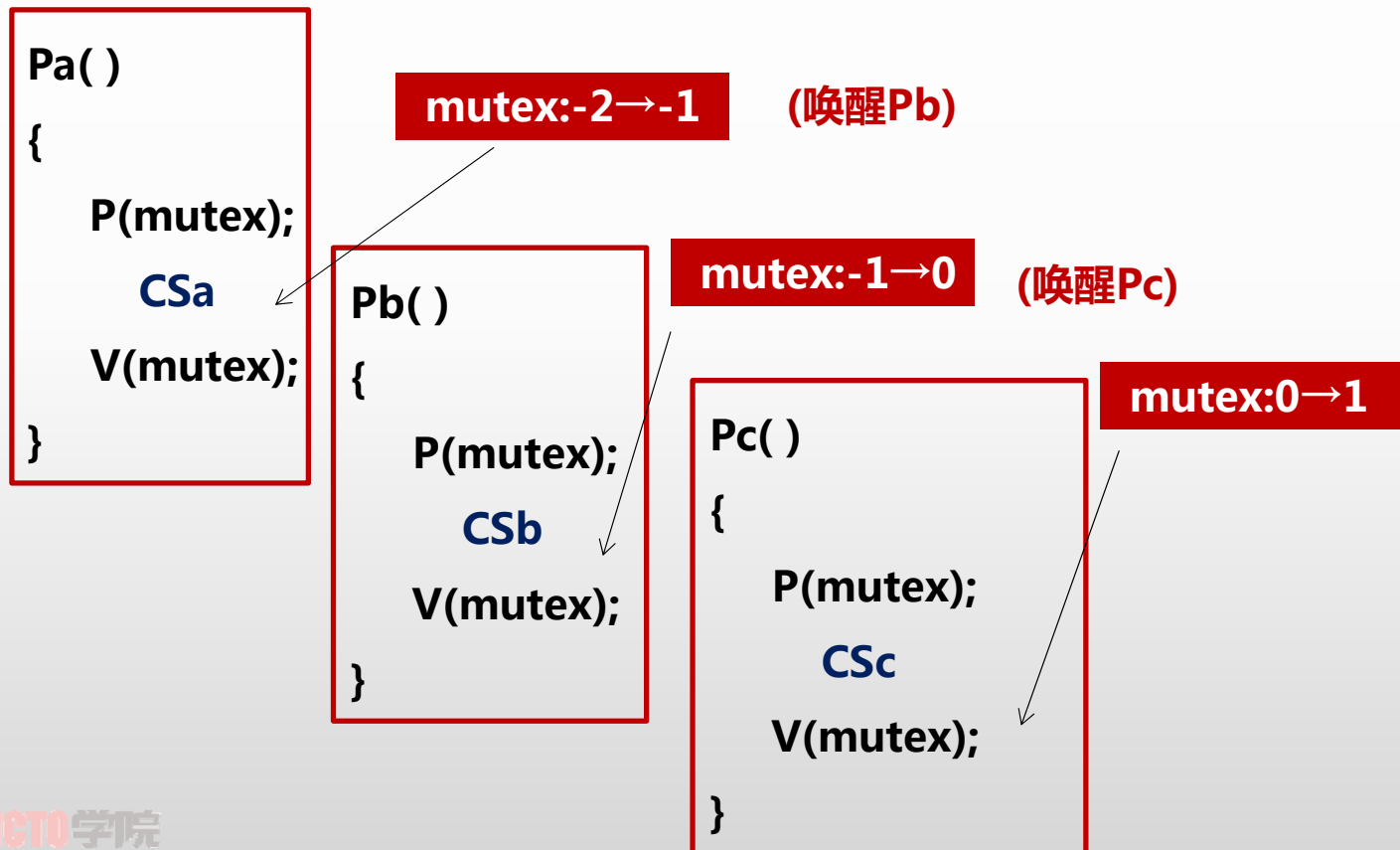
mutex:0→-1

```
Pc( )  
{  
    P(mutex);  
    CSc  
    V(mutex);  
}
```

mutex:-1→-2

PV操作

int mutex=-2;



典型真题

假设系统中有 n 个进程共享3台打印机，任一进程在任一时刻最多只能使用1台打印机。若用PV操作控制 n 个进程使用打印机，则相应信号量 s 的取值范围为（1）；若信号量 S 的值为-3，则系统中有（2）个进程等待使用打印机。

- (1) A. $0, -1, \dots, -(n-1)$ B. $3, 2, 1, 0, -1, \dots, -(n-3)$
C. $1, 0, -1, \dots, -(n-1)$ D. $2, 1, 0, -1, \dots, -(n-2)$

- (2) A. 0 B. 1 C. 2 D. 3

典型真题

试题分析

试题(1)的正确答案为选项B。根据题意，假设系统中有n个进程共享3台打印机，意味着每次只允许3个进程进入互斥段，那么信号量的初值应为3。可见，根据排除法只有选项B中含有3。

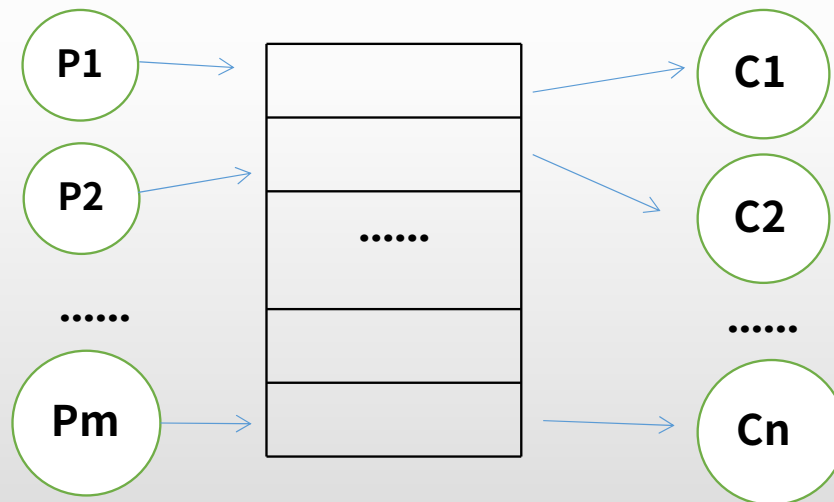
试题(2)的正确答案为选项D。信号量S的物理意义为：当 $S \geq 0$ 时，表示资源的可用数；当 $S < 0$ 时，其绝对值表示等待资源的进程数。

参考答案： (1) B (2) D

PV操作

□ 同步问题：

- 运行条件不满足时，能让进程暂停（在关键操作之前执行P操作）
- 运行条件满足时，能让进程继续（在关键操作之后执行V操作）



规则：

- 不能向**满**缓冲区**存**产品
- 不能向**空**缓冲区**取**产品
- 每个时刻仅允许1个生产者或消费者**存**或**取**1个产品

m 个生产者 能存放5个数据的缓冲区 n 个消费者

PV操作

int empty=5; /*信号量：缓冲区中空位的个数，初值5*/

int full=0; /*信号量：缓冲区中数据的个数，初值0*/

int mutex=1; /*信号量：缓冲区互斥使用，初值1，可用*/

```
producer_i() //i=1...m
{
    while(TRUE)
    {
        生产一个数据;
        P(empty);
        存一个数据到缓冲区;
        V(full);
    }
}
```

```
consumer_j() //j=1...k
{
    while(TRUE)
    {
        P(full);
        从缓冲区取一个数据;
        V(empty);
        消费数据;
    }
}
```

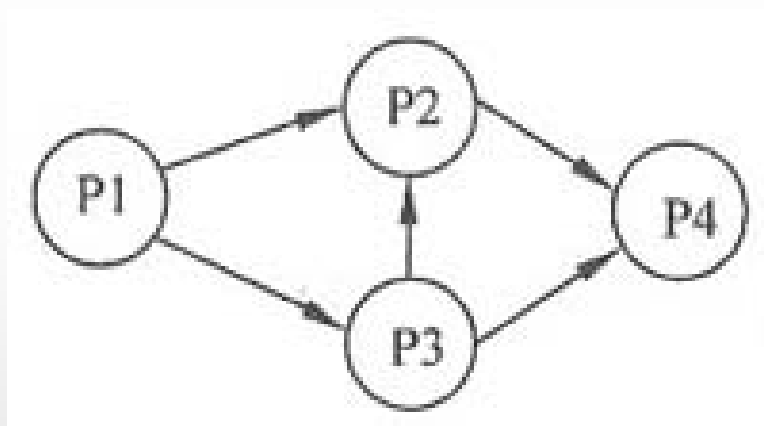
PV操作

```
producer_i( ) //i=1...m
{
    while(TRUE)
    {
        生产一个数据;
        P(empty);
        P(mutex);
        存一个数据到缓冲区;
        V(mutex);
        V(full);
    }
}
```

```
producer_i( ) //i=1...m
{
    while(TRUE)
    {
        生产一个数据;
        P(empty);
        P(mutex);
        从缓冲区取一个数据;
        V(mutex);
        V(full);
        消费数据;
    }
}
```

典型真题

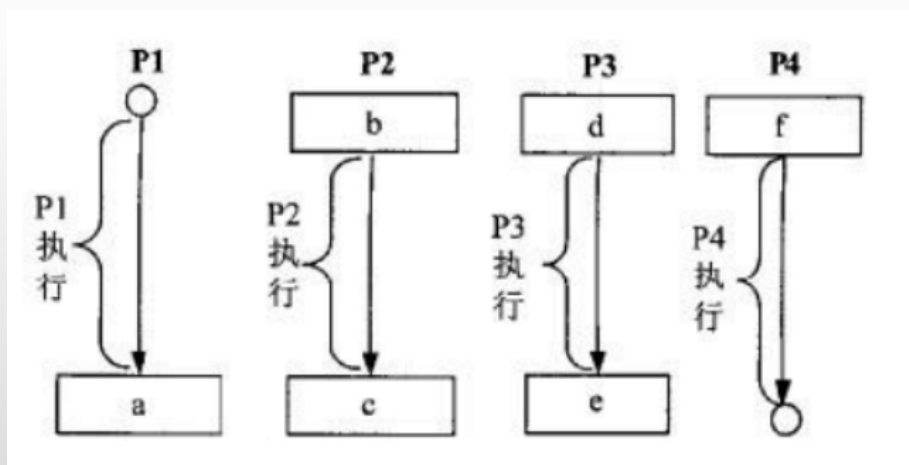
进程P1、P2、P3和P4的前驱图如下所示



若用PV操作控制进程P1~P4并发执行的过程，则需要设置5个信号量S1、S2、S3、S4和S5，且信号量S1~S5的初值都等于0。

典型真题

若用PV操作控制进程P1~P4并发执行的过程，则需要设置5个信号量S1、S2、S3、S4和S5，且信号量S1~S5的初值都等于0。下图中a、b和c处应分别填写（ ）；d、e和f处应分别填写（ ）。



典型真题

- A. $V(S1)V(S2)$ 、 $P(S1)V(S3)$ 和 $V(S4)$
- B. $P(S1)V(S2)$ 、 $P(S1)P(S2)$ 和 $V(S1)$
- C. $V(S1)V(S2)$ 、 $P(S1)P(S3)$ 和 $V(S4)$
- D. $P(S1)P(S2)$ 、 $V(S1)P(S3)$ 和 $V(S2)$

- A. $P(S2)$ 、 $V(S3)V(S5)$ 和 $P(S4)P(S5)$
- B. $V(S2)$ 、 $P(S3)V(S5)$ 和 $V(S4)P(S5)$
- C. $P(S2)$ 、 $V(S3)P(S5)$ 和 $P(S4)V(S5)$
- D. $V(S2)$ 、 $V(S3)P(S5)$ 和 $P(S4)V(S5)$

典型真题

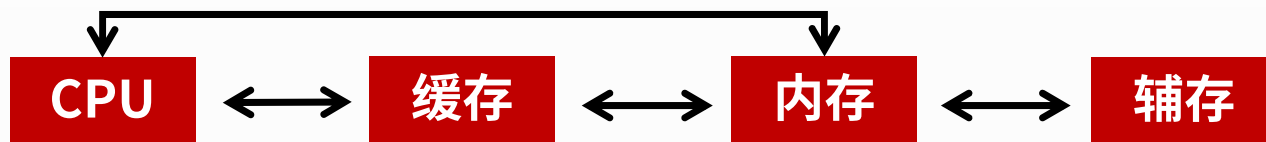
试题分析

因为P1是P2和P3的前驱，当P1执行完需通知P2和P3，应采用V(S1)V(S2)操作分别通知P2和P3，故a处应填写V(S1)V(S2)；又因为P2是P1和P3的后继，当P2执行前应测试P1和P3是否执行完，应采用P(S1)P(S3)操作测试P1和P3是否执行完，故b处应填写P(S1)P(S3)；同理，P2是P4的前驱，当P2执行完应通知P4，应采用V(S4)操作分别通知P4，故c处应填写V(S4)。

因为P3是P1的后继，当P3执行前应测试P1是否执行完，应采用P(S2)操作测试P1是否执行完，故d处应填写P(S2)；又因为P3是P2和P4的前驱，当P3执行完应通知P2和P4，应采用V(S3)V(S5)操作通知P5，故e处应填写V(S3)V(S5)；P4是P2和P3的后继，当P4执行前应测试P2和P3是否执行完，应采用P(S4)P(S5)操作测试P2和P3是否执行完，故f处应填写P(S4)P(S5)。

5 参考答案：C、A

存储管理



- 当内存太小不够用时，用辅存来支援内存
- 暂时不运行的模块换出到辅存上，必要时再换入内存

存储管理

□**地址重定位**是指将程序中的地址虚拟地址（**逻辑地址**）变换成内存的真实地址（**物理地址**）的过程。

逻辑地址

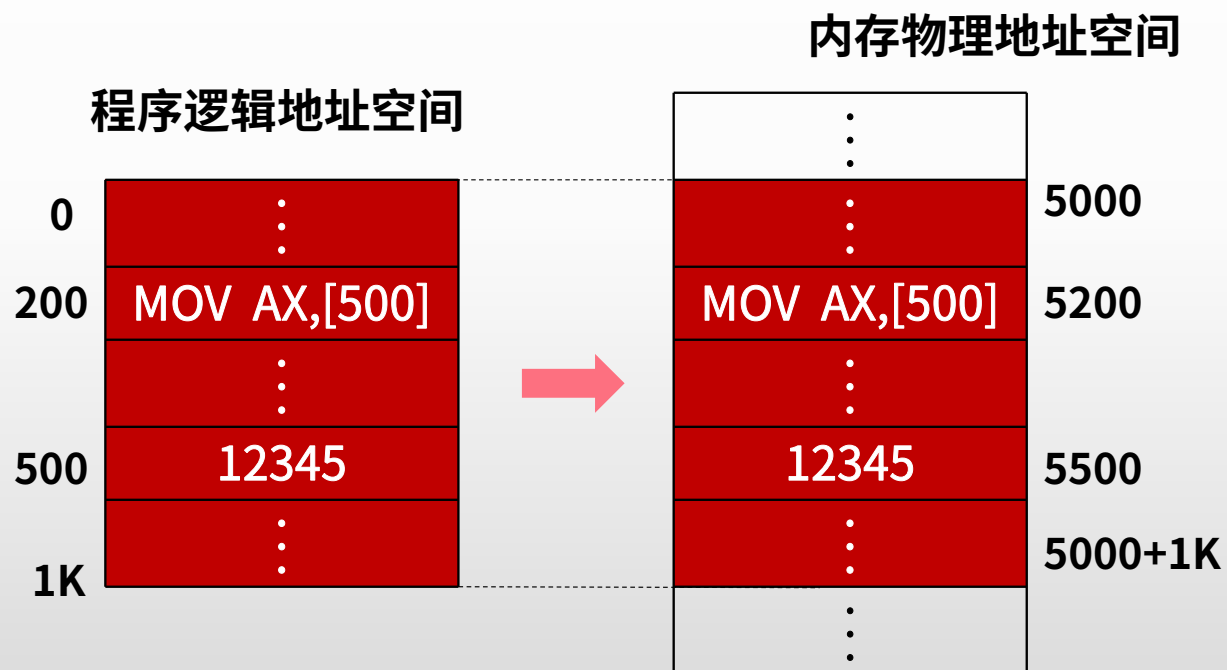
相对地址。
CPU所生成的地址。逻辑地址是内部和编程使用的、并不唯一。

物理地址

绝对地址。
加载到内存地址寄存器中的地址，内存单元的真正地址。

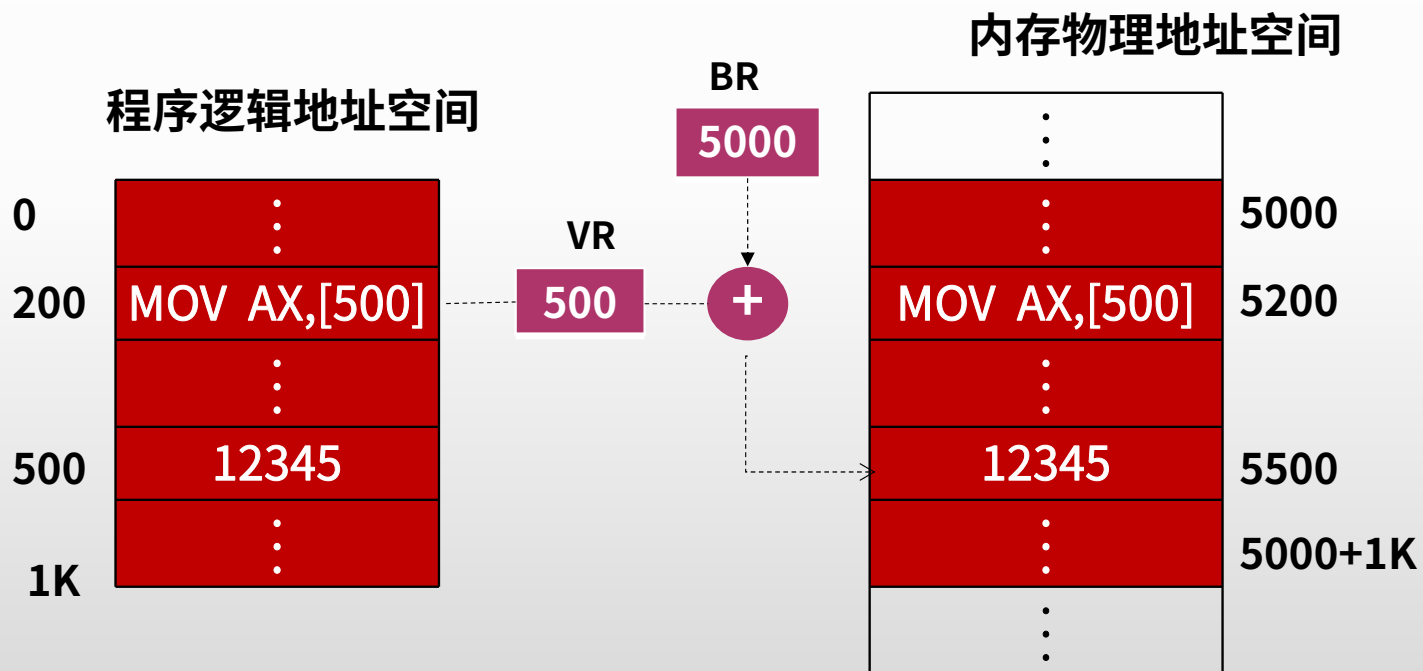
存储管理

□ 静态重定位：绝对地址=相对地址+程序存放的内存起始地址

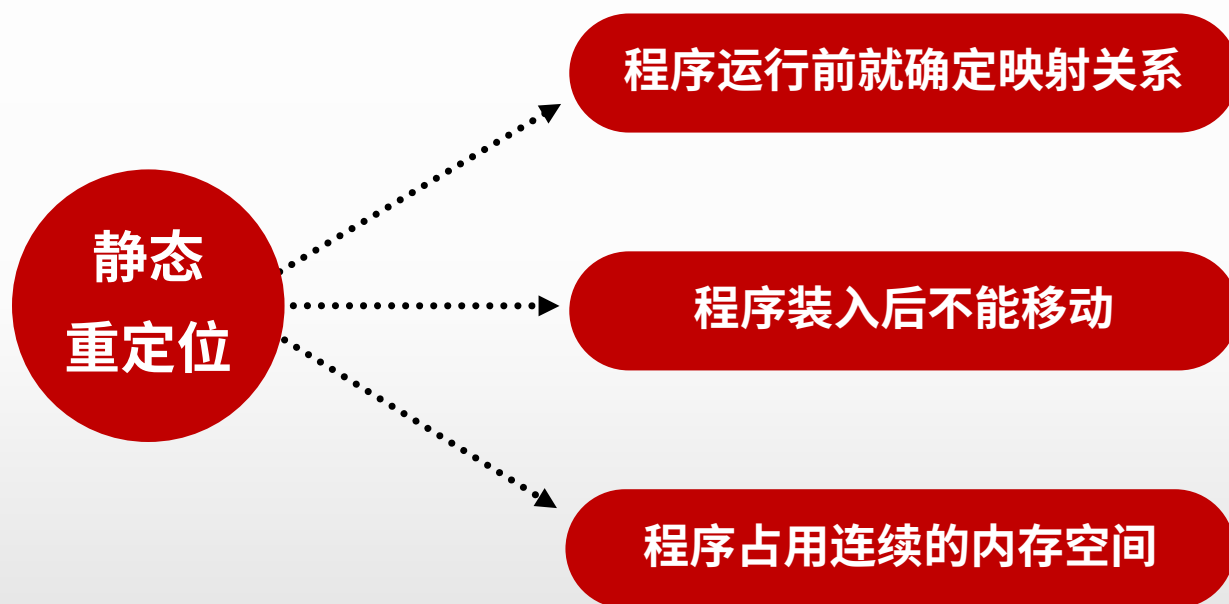


存储管理

□动态重定位：绝对地址=重定位寄存器的值（BR）+逻辑地址寄存器的值（VR）



存储管理

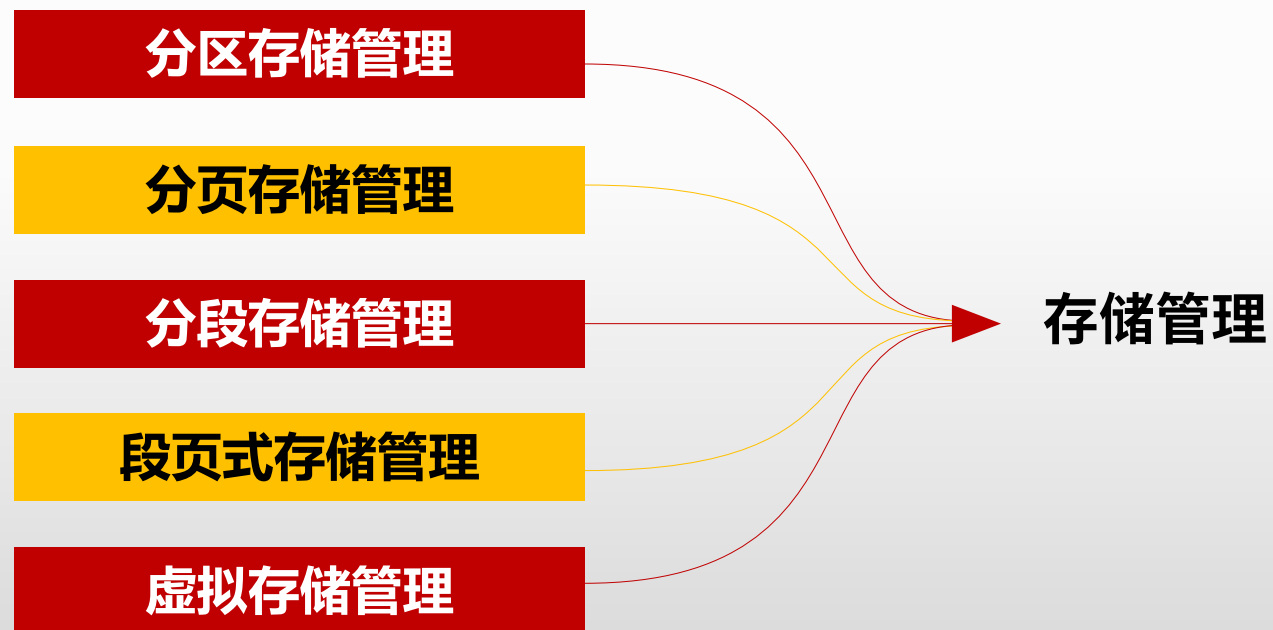


存储管理



存储管理

存储管理的主要目的是解决多个用户使用主存的问题



存储管理

□**分区管理**：把主存的用户区划分成若干个区域，每个区域分配给一个用户作业使用，并限定它们只能在自己的区域中运行。



存储管理

固定分区

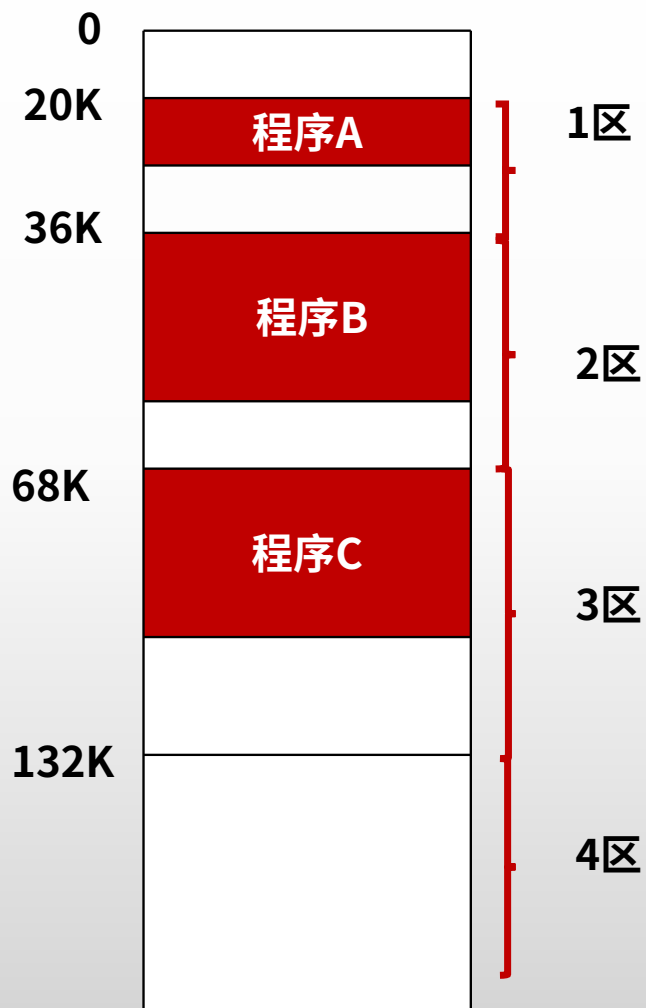
分区表

区号	大小	起始地址	标志
1	16K	20K	已分配
2	32K	36K	已分配
3	64K	68K	已分配
4	124K	132K	未分配

 被占用的空间

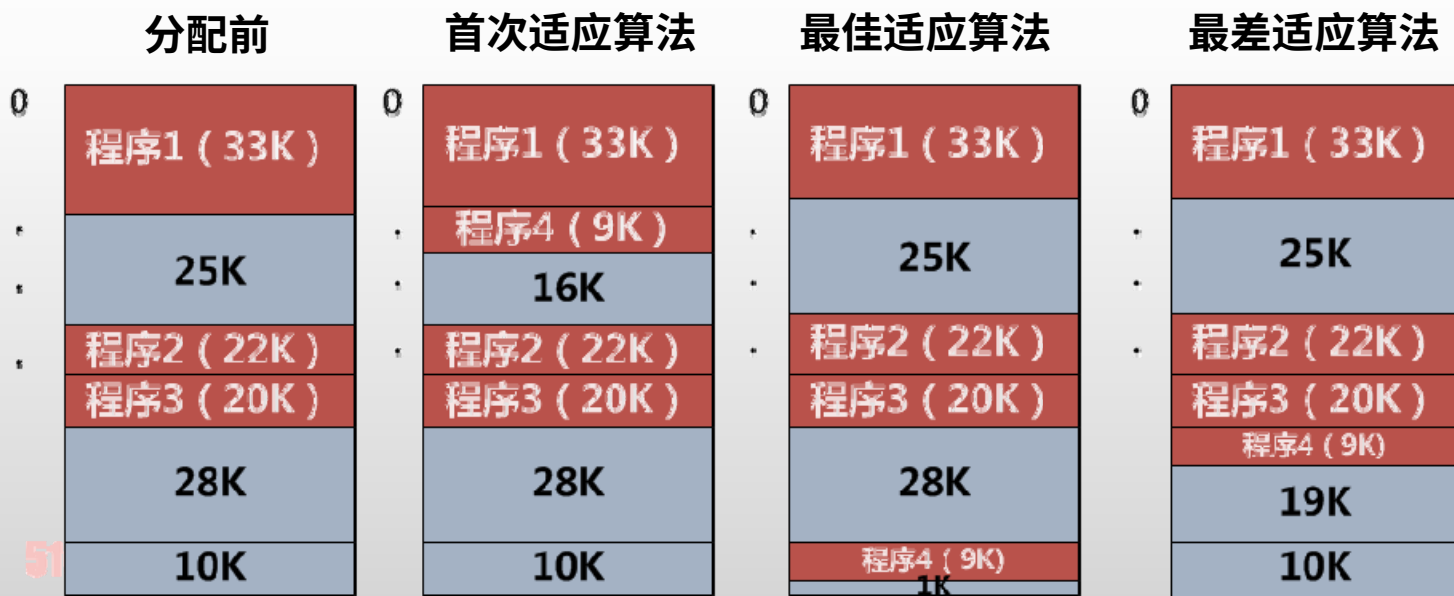
 未占用的空间

51CTO学院



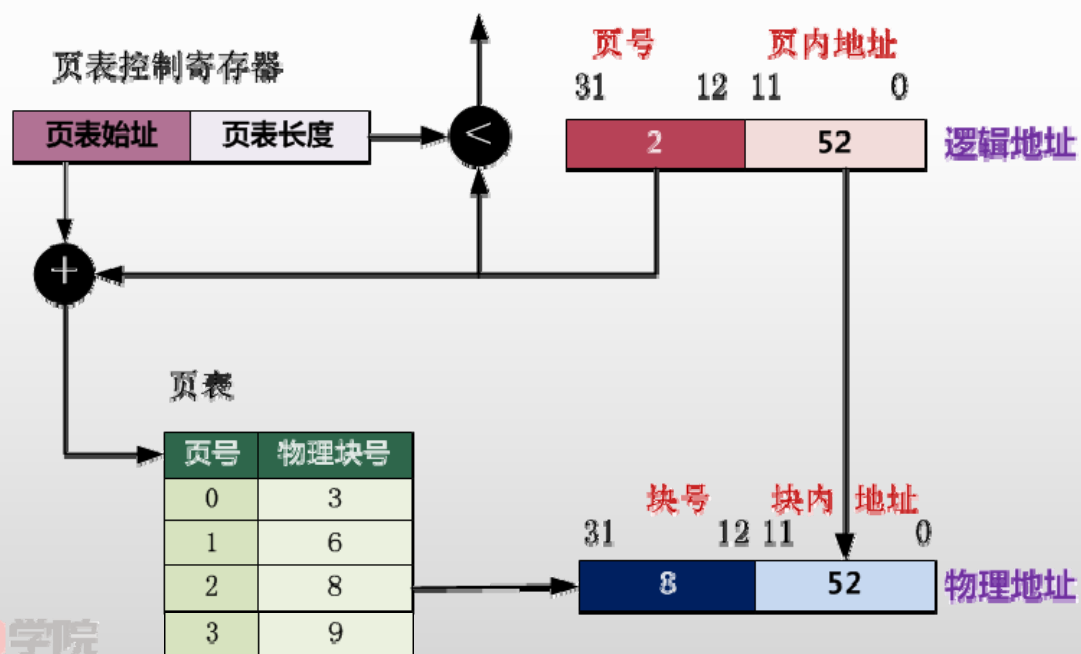
存储管理

可变分区

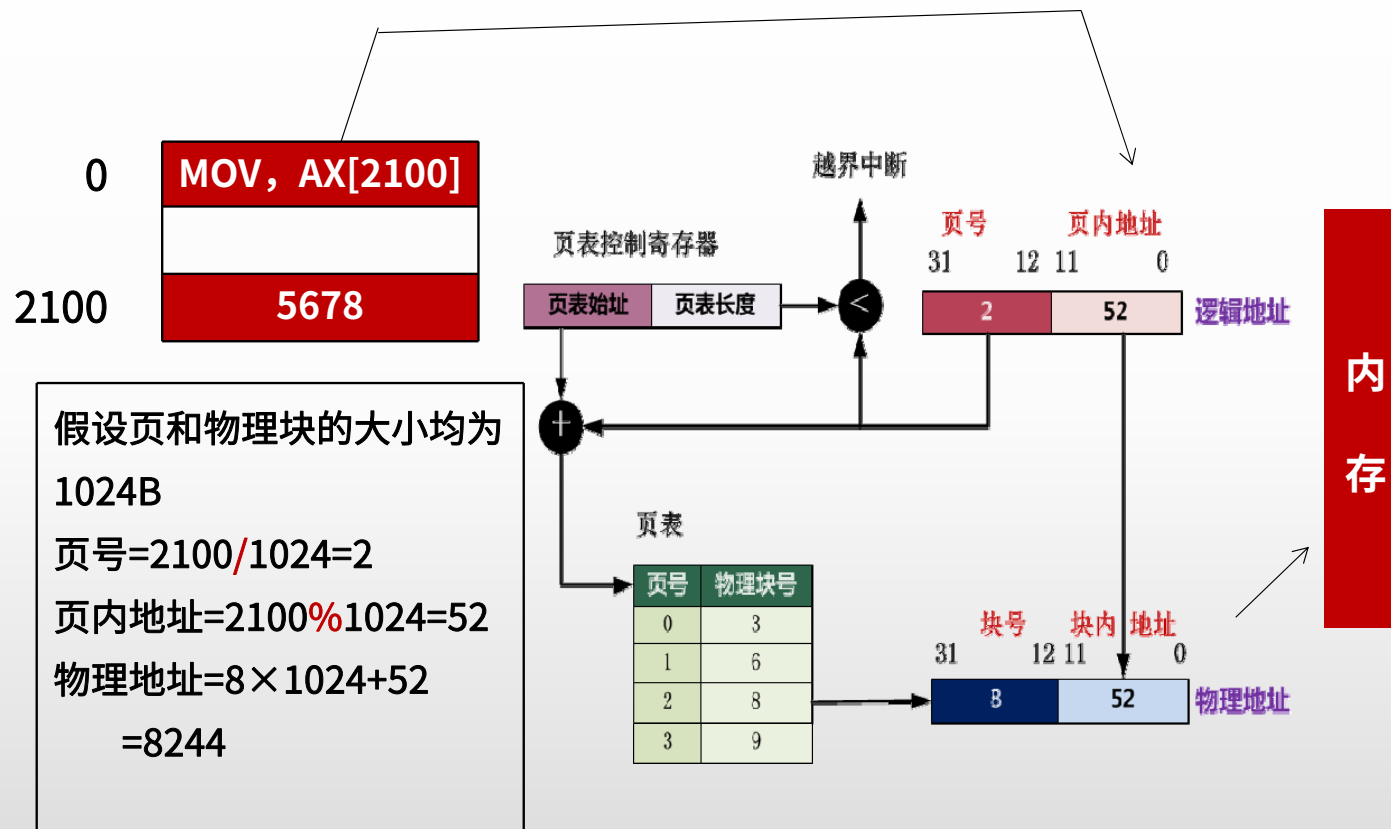


存储管理

- **分页存储管理**：将一个**进程**的地址空间划分成若干个大小相等的区域，称为**页**。相应地，将**主存**空间划分成与页相同大小的若干个物理块，称为**块或页框**。



存储管理



典型真题

某计算机系统页面大小为4K，若进程的页面变换表如下所示，逻辑地址为十六进制1D16H。该地址经过变换后，其物理地址应为十六进制（ ）。

页号	物理块号
0	1
1	3
2	4
3	6

- A. 1024H B. 3D16H C. 4D16H D. 6D16H

典型真题

试题分析

页面大小为4K，而 $4K=2^{12}$ ，因此逻辑地址的低12位对应页内地址，高位对应页号。题目中逻辑地址为十六进制1D16H，一位十六进制数对应4位二进制数，3位十六进制数则对应12位二进制数，因此D16H为页内地址，页号为1。查页面变换表，页号1对应的物理块号为3，将物理块号与页内地址D16H拼接起来即可得到物理地址3D16H。

参考答案： B

典型真题

某操作系统采用分页存储管理方式，下图给出了进程A和进程B的页表结构。如果物理页的大小为512字节，那么进程A逻辑地址为1111(十进制)的变量存放在（ ）号物理内存页中。假设进程A的逻辑页4与进程B的逻辑页5要共享物理页8，那么应该在进程A页表的逻辑页4和进程B页表的逻辑页5对应的物理页处分别填（ ）。

进程A 页表	
逻辑页	物理页
0	9
1	2
2	4
3	6
4	
5	

进程B 页表	
逻辑页	物理页
0	1
1	3
2	5
3	7
4	2
5	

物理页
0
1
2
3
4
5
6
7
8
9

典型真题

A. 9 B. 2 C. 4 D. 6

A. 4、5 B. 5、4 C. 5、8 D. 8、8

典型真题

试题分析

十进制1111转化为二进制10001010111。物理页的大小为512字节,这说明页内地址为9个二进制位,进程A的逻辑址中,右边的9位是页内地址,左边的2位是页号,即:10 001010111。页号为二进制的10.即十进制的2,对应的物理页号为4.

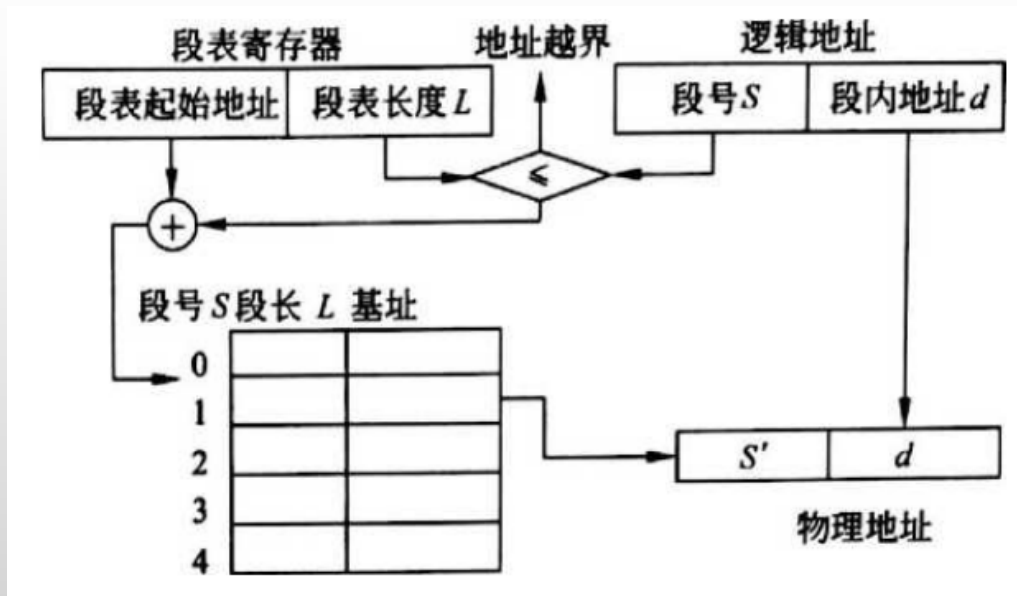
若A页表的透辑页4和进程B页表的透辑页5共享物理页8,则说明他们都对应物理页8.所以均填8。

参考答案: C、D

存储管理

分段管理

分段式存储管理系统中，为每个段分配一个连续的分区，而进程中的各个段可以离散地分配到主存的不同分区中。在系统中为每个进程建立一张段映射表，简称为“段表”。每个段在表中占有一个表项，在其中记录了该段在主存中的起始地址（又称为“基址”）和段的长度。进程在执行时，通过查段表来找到每个段所对应的主存区。



典型真题

假设系统采用段式存储管理方法，进程P的段表如下所示。逻辑地址（ ）不能转换为对应的物理地址；不能转换为对应的物理地址的原因是进行（ ）。

段号	基地址	段长
0	1100	800
1	3310	50
2	5000	200
3	4100	580
4	2000	100

典型真题

- A. (0, 790)和(2, 88) B. (1, 30)和(3, 290)
C. (2, 88)和(4, 98) D. (0, 810)和(4, 120)

- A. 除法运算时除数为零
B. 算术运算时有溢出
C. 逻辑地址到物理地址转换时地址越界
D. 物理地址到逻辑地址转换时地址越界

典型真题

试题分析

给定段地址(x, y)，其中：x为段号，y为段内地址。将(x, y)转换为物理地址的方法是：根据段号x查段表→判断 $y < \text{段长}$ ；如果小于段长，则物理地址=基地址+段内地址y，否则地址越界。

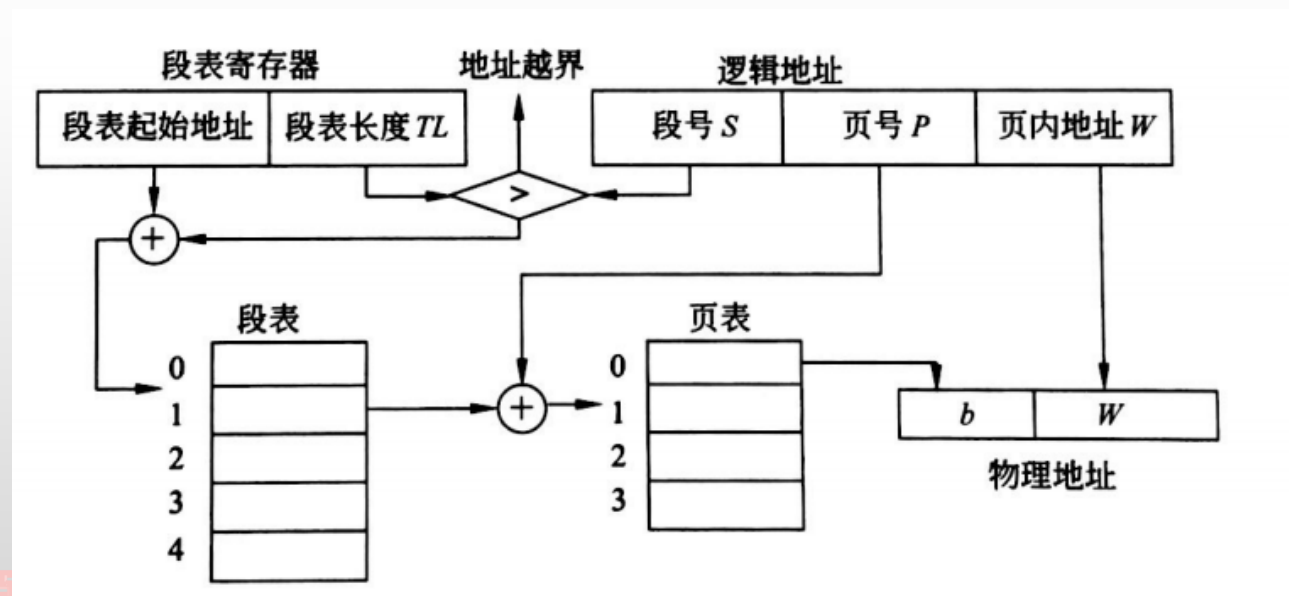
因为段地址(0, 810)中，0段的段长为800，段内地址810大于段长，故地址越界。段地址(4, 120)中，4段的段长为100，段内地址120大于段长，故地址越界。

参考答案：D、C

存储管理

段页式管理

段页式系统的基本原理是先将整个主存划分成大小相等的存储块（页框），将用户程序按程序的逻辑关系分为若干个段，再将每个段划分成若干页，以页框为单位离散分配。在段页式系统中，其地址结构由段号、段内页号和页内地址三部分组成。



存储管理

虚拟存储

在前面介绍的存储管理方案中，必须为每个作业分配足够的空间，以便装入全部信息。当主存空间不能满足作业要求时，作业无法装入主存执行。如果一个作业只部分装入主存便可开始启动运行，其余部分暂时留在磁盘上，在需要时再装入主存，这样可以有效地利用主存空间。从用户角度看，该系统所具有的主存容量将比实际主存容量大得多，人们把这样的存储器称为虚拟存储器。

1

请求分页存储

2

请求分段存储

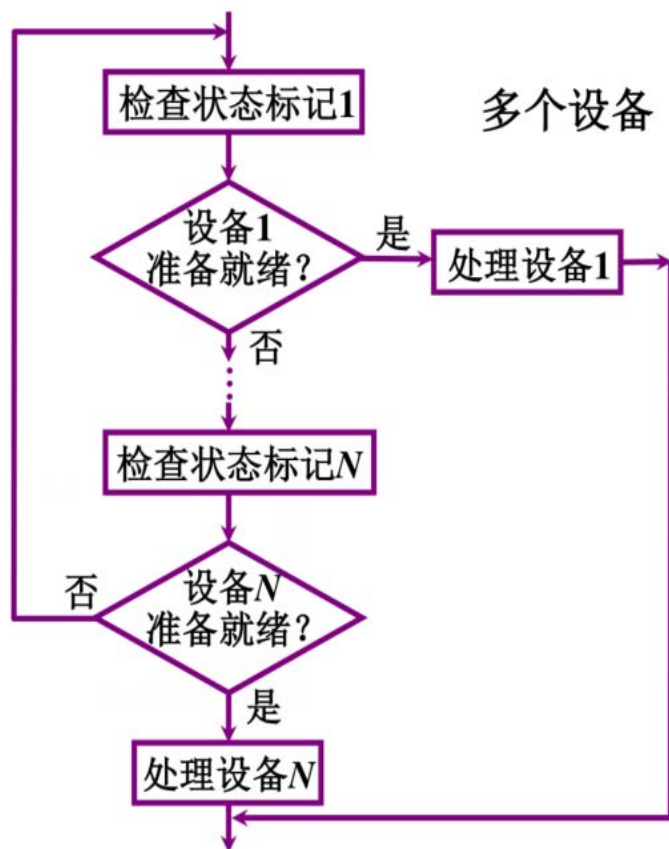
3

请求段页式存储

设备管理

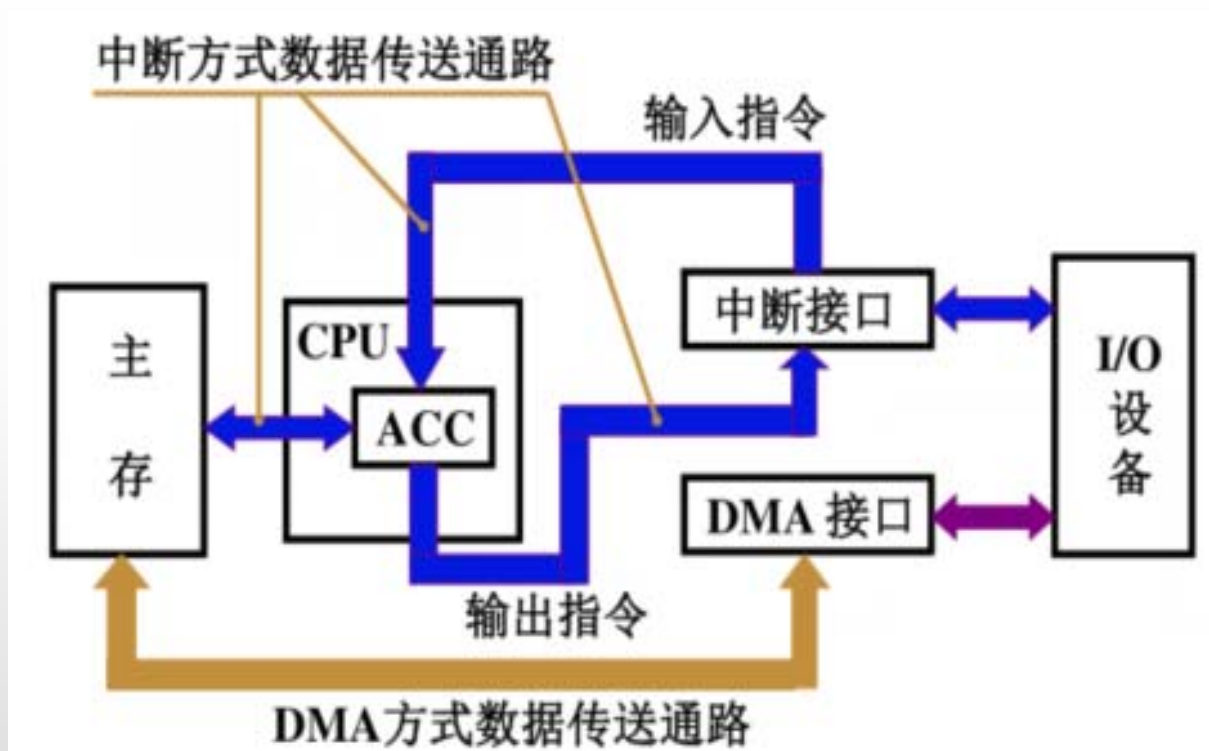
工作方式	特点	
程序控制	无条件传送	I/O端口总是准备好，cpu在需要时，随时直接利用访问相应的I/O端口。
	程序查询	CPU必须不停地测试I/O设备的状态端口。CPU与I/O设备是串行工作的。
中断	某个进程要启动某个设备时，CPU就向相应的设备控制器发出一条设备I/O启动指令，然后CPU又返回做原来的工作。CPU与I/O设备可以并行工作。	
DMA(直接内存存取)	通过DMA控制器直接进行批量数据交换，除了在数据传输开始和结束时，整个过程无须CPU的干预。	

设备管理



中断方式

设备管理



典型真题

DMA (直接存储器访问) 工作方式是在 () 之间建立起直接的数据通路。

- A. CPU 与外设
- B. CPU 与主存
- C. 主存与外设
- D. 外设与外设

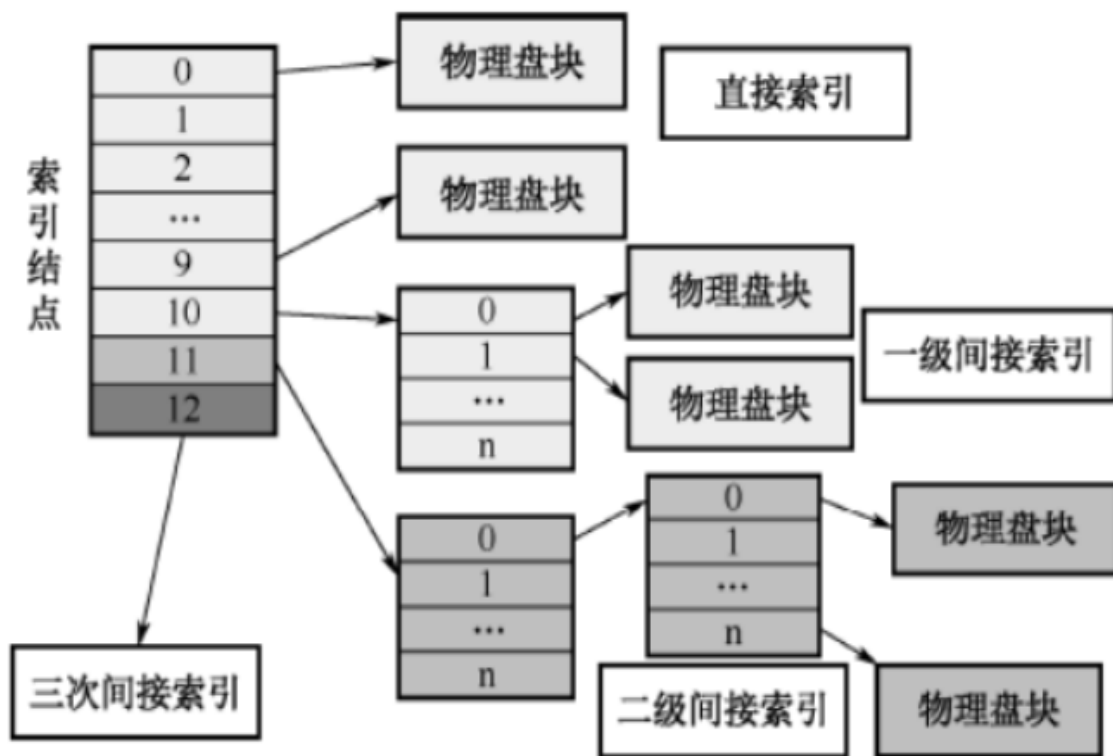
典型真题

试题分析

直接主存存取（Direct Memory Access, DMA）是指数据在主存与I/O设备间的直接成块传送，即在主存与I/O设备间传送数据块的过程中，不需要CPU作任何干涉，只需在过程开始启动（即向设备发出“传送一块数据”的命令）与过程结束（CPU通过轮询或中断得知过程是否结束和下次操作是否准备就绪）时由CPU进行处理，实际操作由DMA硬件直接完成，CPU在传送过程中可做其它事情。

参考答案：C

文件存储管理



典型真题

● 某文件系统文件存储采用文件索引节点法。假设文件索引节点中有8个地址项*iaddr*[0]~*iaddr*[7]，每个地址项大小为4字节，其中地址项*iaddr*[0]~*iaddr*[5]为直接地址索引，*iaddr*[6]是一级间接地址索引，*iaddr*[7]是二级间接地址索引，磁盘索引块和磁盘数据块大小均为4KB。该文件系统可表示的单个文件最大长度是（ ）KB。若要访问*iclsClient.dll*文件的逻辑块号分别为6、520和1030，则系统应分别采用（ ）。

A. 1030 B. 65796 C. 1049606 D. 4198424

A. 直接地址索引、一级间接地址索引和二级间接地址索引

B. 直接地址索引、二级间接地址索引和二级间接地址索引

C. 一级间接地址索引、一级间接地址索引和二级间接地址索引

D. 一级间接地址索引、二级间接地址索引和二级间接地址索引

典型真题

试题解析：

直接索引范围： $6 \times 4\text{KB} = 24\text{KB}$ ，对应逻辑块号：0-5；

一级间接索引范围： $(4\text{KB}/4\text{B}) \times 4\text{KB} = 4096\text{KB}$ ，对应逻辑块号：6-1029；

二级间接索引范围： $(4\text{KB}/4\text{B}) \times (4\text{KB}/4\text{B}) \times 4\text{KB} = 4194304\text{KB}$ ，
对应逻辑块号：1030以及上。

单个文件最大长度是： $24\text{KB} + 4096\text{KB} + 4194304\text{KB} = 4198424\text{KB}$

参考答案：D、C

文件存储设备管理

位示图法。该方法是在外存上建立一张位示图（Bitmap），记录文件存储器的使用情况。每一位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。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
第1个字	1	1	0	0	0	1	1	1	0	0	1	0	1	1	1	0
第2个字	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
第3个字	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
第4个字																
⋮																
第15个字																

典型真题

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。若磁盘上物理块的编号依次为0、1、2、....。系统中的字长为64位，字的编号依次为0、1、2、...。字中的一位对应文件存储器上的一个物理块。取值0和1分别表示空闲和占用。如下图所示。假设操作系统将256号物理块分配给某文件，那么该物理块的使用情况在位示图中编号为()的字中描述，系统应该将()。

字号								位号							
63 62 ...				3 2		1	0								
0	1	...		1	0	0	0	1							
1	1	...		1	0	1	1	0							
0	1	...		0	1	1	0	1							
0	1	...		1	0	1	0	1							
		...													
1	1	...		0	1	0	0	1							

典型真题

- A.3 B.4
- C.5 D.6

- A.该字的0号位置 “1”
- B.该字的63号位置 “1”
- C.该字的0号位置 “0”
- D.该字的63号位置 “0”

典型真题

试题解析：

题考查位示图知识。注意：此题的字号与位号均从0开始。

由于物理块从0开始，从0块到255块刚好占用了4个字（ $64*4=256$ ），256块应该是第五个字（4号字）的0号位置。

参考答案：B、A

技术成就梦想