

# PPT章末习题

2021年12月4日

# 第一章引论——小结

熟悉和掌握以下基本概念：

操作系统、批处理技术、批处理系统、分时系统、  
实时系统、脱机输入输出、多道程序设计技术

熟悉和掌握以下基本知识：

1. 操作系统的目标
2. 操作系统的作用
3. 操作系统的特征
4. 操作系统的基本功能

# 本章习题

(1) 当CPU执行操作系统代码时, 称处理机处于( **C** )

A. 执行态      B. 目态      C. 管态      D. 就绪态

(2) 在下列性质中, ( **D** )不是分时系统的特征。

A. 多路性      B. 交互性      C. 独立性      D. 成批性

1、在计算机系统中配置操作系统的主要目的是( ),  
操作系统的主要功能是管理计算机系统( ),  
其中包括( )管理, ( )管  
理以及设备管理和文件管理, 这里的( )管  
理主要是对进程进行管理。

答:提高系统资源的利用率

资源

处理机

存储器

处理机

2、利用缓冲区能有效地缓和( )和( )之  
间速度不匹配的矛盾, 虚拟设备的功能是使( )变  
成能被多个进程同时使用的( )。

答: CPU

I/O设备

一个物理设备

逻辑设备

# 第二章进程管理——小结

本章是本教材的重点和难点, 主要讲述:

- 程序并发执行及特点、进程的定义、进程的状态及转换、进程控制块及作用;
- 进程的互斥与同步及其区别与联系, 并着重介绍运用操作系统提供的同步机制实现进程间的互斥与同步, 解决经典的进程同步问题;
- 讨论了进程间的通信;
- 掌握进程与线程区别和联系;

# 习题

- 1、 对一个可执行程序文件, 该程序与执行它的进程是 ( 一对多 ) 的关系。
- 2、 在单CPU系统中实现并发技术后 ( A )
  - A. 进程间在一个时间段内同时执行, CPU与外设并行工作。
  - B. 进程间在一个时刻同时执行, CPU与外设并行工作。
  - C. 进程间在一个时间段内同时执行, CPU与外设串行工作。
  - D. 进程间在一个时刻同时执行, CPU与外设串行工作。
- 3、 从静态角度上看, 进程是由 ( PCB(或进程控制块) )、 ( 程序段 )、 ( 数据段 ) 三部分组成。
- 4、 正在执行的进程由于用完其时间片而被暂停执行, 此时进程应从执行状态变成为 ( 就绪 ) 状态。

- 5、引入进程,可带来( 资源利用率的提高 )和( 系统吞吐量的增加 )的好处,但却增加了系统的( 时间 )和( 空间 )开销。
- 6、临界区是指进程中用于( 访问临界资源 )的那段代码。
- 7、 C 是一种只能由P和V操作所改变的整型变量, (1)可用于实现进程的 (2) D (3) A (2)是指排他性地访问临界资源。
- (1) A. 控制变量      B. 锁      C. 整型信号量      D. 记录型信号量
- (2) (3) A. 同步      B. 通信      C. 调度      D. 互斥
- 8、设有6个进程共享同一互斥段,若最多允许有3个进程进入互斥段,则所采用的信号量的初值为( 3 )。
- 9、有3个进程共享同一程序段,而每次最多允许两个进程进入该程序段,若用P、V操作作同步机制,则记录型信号量S的取值范围为( 2,1,0,-1(或[2,-1]) )。



10、为实现消息缓冲队列通信,在PCB中应增加 **消息队列首指针** **消息队列互斥信号量** **消息队列资源信号量** )三个数据项。

11、若记录型信号量S的初值为2,当前值为-1,则表示有( **B** )等待进程。

A. 0个                      B. 1个                      C. 2个                      D. 3个

12、当( **B** )时,进程从执行状态转变为就绪状态。

A. 进程被调度程序选中                      B. 有高优先级进程来  
C. 等待某一事件                      D. 等待的事件发生

13、在进程状态转换时,下列( **D** )转换是**不可能**发生的。

A. 就绪态—>执行态                      B. 执行态—>就绪态  
C. 执行态—>阻塞态                      D. 阻塞态—>执行态

14、下列各项工作步骤中, ( B ) 不是创建进程所必需的步骤。

- A. 申请一个PCB
- B. 阻塞进程
- C. 为进程分配内存等必要资源
- D. 将PCB连接入进程就绪队列

15、在操作系统中, 死锁出现指的是 ( C )

- A. 计算机发生了重大故障
- B. 资源数远远少于进程数
- C. 若干进程因竞争资源而无限等待其他进程释放已占有的资源
- D. 进程同时申请的资源数超过资源总数。

16、下列关于进程的叙述中, 正确的是 ( A )

- A. 进程获得CPU而运行是通过调度得到的。
- B. 优先级是进行进程调度的重要依据, 一旦确定不能改变。
- C. 在单CPU系统中, 任一时刻都有1个进程处于运行状态。
- D. 进程申请CPU得不到满足时, 其状态变为等待状态。

17、在直接通信方式中,系统通常提供的两条通信原语如下,请选择适当的参数填入。

send(   B  ,   D   ); receive(   A  ,   D   )

A. Sender    B. receiver    C. text    D. message    E. mailbox

18、下列原语中,能进行进程通信的原语是(   C   )和(   D   )

A. 挂起/撤销    B. 创建/撤销    C. P、V操作    D. 锁/开锁

19、消息缓冲队列通信中的临界资源是(   B   )

A. 队列中的某个消息缓冲区    B. 整个消息缓冲队列  
C. 信箱    D. 管道

20、处于执行状态中的进程若同时发生了下列两种情况:(a)对某信号量执行P操作后,其结果为负。(b)时间片到了中断发生。则该进程将由执行状态变迁为(   A   )状态。

A. 阻塞    B. 就绪    C. 阻塞或就绪    D. 不定

- 1、设有无穷多个缓冲区和无穷多个信息, A进程把信息逐个地写入每个缓冲区, B进程则逐个地从缓冲区中取出信息。
- 试问: (1) 两个进程之间的制约关系。
- (2) 用P、V操作写出两进程的同步算法, 并给出信号量的初值。
- (3) 指出信号量的变化范围和其值的含义。

答: (1) **B进程不能超前A进程, 但A进程不受B进程的约束。**

(2) var

S: semaphore;

**S:=0;**

begin

parbegin

process A

begin

i:=0;

L: i:=i+1;

写入第i个缓冲区;

**V(S);**

goto L;

end;

parend;

end;

process B

begin

j:=0;

M: j:=j+1;

**P(S);**

取出第j个缓冲区信息;

goto M;

end;

(3) 信号量S的值域为 $[-1, \infty]$ 中的整数, 当 $S=-1$ 时, 表示缓冲区没有信息(或B读空), 且B要求进一步读出, 也即超前A欲读取信息而受阻。

2、购物问题。某超级市场,可容纳100人同时购物,入口处备有篮子,每个购物者可持一个篮子入内购物,出口处结帐,并归还篮子(出、入口(一共2个口)仅容纳一人通过),请用P、V操作完成购物同步算法。

```
答:var
    S, mutex1, mutex2: semaphore;
    S:=100;
    mutex1:=1;
    mutex2:=1;
begin
    parbegin
        process Pi
            begin
                P(S);
                P(mutex1);
                进口处, 取一只篮子;
                V(mutex1);
                选购商品;
                P(mutex2);
                结帐, 并归还篮子;
                V(mutex2);
                V(S);
            end;
        parend;
    end;
```

# 独木桥问题

3、某条河上只有一座独木桥(东西向),以便行人过河。现在河的两边都有人要过桥,按照下面的规则过桥,为了保证过桥安全,请用P、V操作分别实现正确的管理。

**规则:** (1) 每次只有一个人通过桥。

(2) 同一方向的可连续过桥,某方向有人过桥时另一方向的人要等待。



```

(1)
var
    mutex: semaphore := 1;
begin
    parbegin
        Process (E-W)i (i=1, 2, ...)
            begin
                P(mutex);
                过桥;
                V(mutex);
            end;
    parend;
end;

Process (W-E)j (j=1, 2, ...)
    begin
        P(mutex);
        过桥;
        V(mutex);
    end;

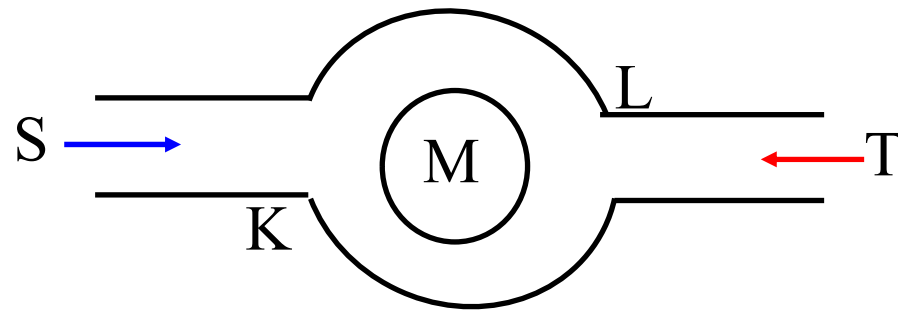
```

```

(2)
var
    S, S1, S2: semaphore:=1, 1, 1;
    rc1, rc2: integer:=0, 0;
begin
    parbegin
        Process (E-W)i (i=1, 2, ...)
            begin
                P(S1);
                rc1:=rc1+1;
                if rc1=1 then P(S);
                V(S1);
                过桥;
                P(S1);
                rc1:=rc1-1;
                if rc1=0 then V(S);
                V(S1);
            end;
        Process (W-E)j (j=1, 2, ...)
            begin
                P(S2);
                rc2:=rc2+1;
                if rc2=1 then P(S);
                V(S2);
                过桥;
                P(S2);
                rc2:=rc2-1;
                if rc2=0 then V(S);
                V(S2);
            end;
    parend;
end;

```

4、小路问题。在两地之间有一条弯曲小路, 其中S到T的一段路每次只允许一辆自行车通过, 但中间有一个小的“安全岛” M(同时允许两辆自行车停留), 可供两辆自行车在从两端进入小路情况下错车使用, 如图, 试设计一个算法使来往的自行车均可顺利通过。(同方向不允许连续过自行车)



答:var

S, SK, T, TL: semaphore:=1, 1, 1, 1;

begin

parbegin

Process ST

begin

P(S);

P(SK);

通过SK;

V(SK);

进入M;

P(TL);

通过LT;

V(TL);

V(S);

end;

parend;

end;

Process TS

begin

P(T);

P(TL);

通过TL;

V(TL);

进入M;

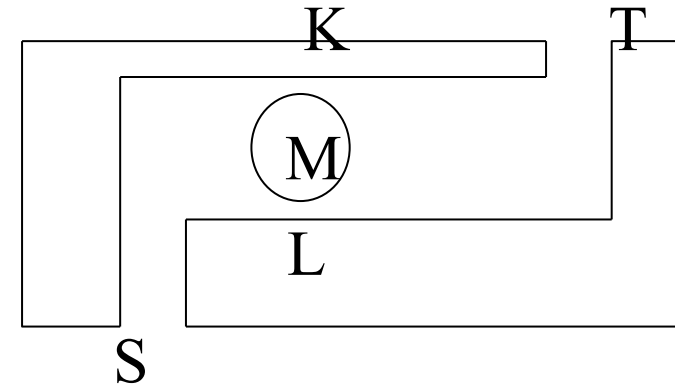
P(SK);

通过KS;

V(SK);

V(T);

end;



5、拣棋子问题。生产围棋的工人不小心把相等数量的黑棋子和白棋子混装在一个箱子里,现要用自动分拣系统把黑棋子和白棋子分开,该系统由两个并发执行的进程组成,系统功能如下:

- (1) 进程A专门拣黑子, 进程B专门拣白子;
- (2) 每个进程每次只拣一个, 当一个进程在拣子时, 不允许另一个进程去拣子;
- (3) 当一个进程拣了一个子(黑或白)以后, 必让另一个进程拣一个子(白或黑);
- (4) 进程A先执行。

```

答: var
      S1, S2: semaphore:=1, 0;
begin
  paerbegin
    Process A
      begin
        repeat
          P(S1);
          拣一个黑子;
          V(S2);
        until false;
      end;
    parend;
  end;
  Process B
    begin
      repeat
        P(S2);
        拣一个白子;
        V(S1);
      until false;
    end;
  end;
end;

```

6、某寺庙有小、老和尚若干,有一个水缸,由小和尚提水入水缸供老和尚饮用。水缸可以容纳10桶水,水取自同一井水。水井狭窄,每次只能容一个桶取水。水桶总数为3个,每次入、出水缸仅一桶,且不可同时进行。试P、V操作描述算法。(提示:老和尚从水缸取水需要用桶)

```

答: var
    mutex1, mutex2, empty, full, count: semaphore:=1, 1, 10, 0, 3;
begin
    parbegin
        processs 小和尚
            begin
                repeat
                    P(empty);
                    P(count);
                    P(mutex1);
                    从井中取水;
                    V(mutex1);
                    P(mutex2);
                    送水入水缸;
                    V(mutex2);
                    V(count);
                    V(full);
                until false;
            end;
        parend;
    end;
end;

        processs 老和尚
            begin
                repeat
                    P(full);
                    P(count);
                    P(mutex2);
                    从缸中取水;
                    V(mutex2);
                    V(count);
                    V(empty);
                until false;
            end;

```



# 第三章处理机调度与死锁——小结

- 处理器管理的主要任务是分配处理器
- 主要目的是提高处理器的使用效率。
- **主要内容**: 三大调度(高级、中级、低级)、调度模型、**调度算法**(3个)、**实时调度**、**多处理机系统中的调度**、死锁和死锁处理(**银行家算法**)

# 处理死锁的基本方法

| 方法                      | 资源分配策略                             | 各种可能模式               | 主要优点                           | 主要缺点                                     |
|-------------------------|------------------------------------|----------------------|--------------------------------|--|
| <b>预防</b><br>Prevention | 保守的；宁可资源闲置（从机制上使死锁条件不成立，即摒弃三个必要条件） | 一次请求所有资源<条件2>        | 适用于作突发式处理的进程；不必剥夺              | 效率低；进程开始时间可能延长<br>资源浪费有可能严重<br>不便灵活申请新资源 |
|                         |                                    | 资源剥夺<条件3>            | 适用于状态可以保存和恢复的资源                |  |
|                         |                                    | 资源 <b>按序申请</b> <条件4> | 可以在编译时（而不必在运行时）就进行检查           |  |
| <b>避免</b><br>Avoidance  | 是“预防”和“检测”的折衷（在运行时判断是否可能死锁）        | 寻找可能的安全的运行顺序         | 不必进行剥夺                         | 使用条件：必须知道将来的资源需求；进程可能会长时间阻塞              |
| <b>检测</b><br>Detection  | 宽松的；只要允许，就分配资源                     | 定期检查死锁是否已经发生         | 进程开始时间 <b>不</b> 延长；允许对死锁进行现场处理 | 通过剥夺解除死锁，造成损失                            |

- 1、在批处理、分时和实时操作系统中,都设置了 **C** ( ),在批处理系统中还应设置 **B** ( )。
- A. 剥夺调度 B. 作业调度 C. 进程调度 D. 中级调度
- 2、如果为每一个作业,只建立一个进程,则为了照顾短作业用户,应采用 ( **B** ),为照顾紧急作业的用户,应采用 ( **E** ),为能实现人机交互作用,应采用 ( **C** ),而能使短作业,长作业及交互作业用户都比较满意时,应采用 (**D**)。
- A. FCFS调度算法 B. 短作业优先调度算法  
C. 时间片轮转法 D. 多级反馈队列调度算法  
E. 基于优先权调度算法

3、产生死锁的基本原因是((1) **B**)和((2) **A**),产生死锁的四个必要条件是互斥条件,((3) **C**),不剥夺条件和((4) **B**)。

- |                 |            |
|-----------------|------------|
| (1) A. 资源分配不当   | B. 竞争资源    |
| C. 作业调度不当       | D. 资源的独占性  |
| (2) A. 进程推进顺序不当 | B. 进程调度不当  |
| C. 系统中进程太多      | D. CPU运行太快 |
| (3) A. 请求和阻塞    | B. 请求释放    |
| C. 请求和保持条件      | D. 释放和阻塞条件 |
| (4) A. 线性增长条件   | B. 环路等待条件  |
| C. 无序释放条件       | D. 有序请求条件  |

4、实际操作系统要兼顾资源的使用和安全可靠,对资源的分配策略,往往采用( **D** )策略。

A. 死锁的预防

B. 死锁的避免

C. 死锁的检测

D. 死锁的预防、避免和检测的混合

5、在下列死锁的解决办法中,属于死锁预防策略的是( **B** )

A. 银行家算法

B. 资源有序分配法

C. 死锁检测法

D. 资源分配图化简法

6、资源的一次分配法和有序分配法分别破坏了产生死锁的必要条件中的( **请求和保持条件** )和( **环路等待条件** ),它们属于( **死锁的预防** )。而银行家算法属于( **死锁避免** )。

- 7、作业调度是从( **后备作业队列** )中选出一批作业, 为它们分配( **资源** ), 并为它们创建( **进程** )。
- 8、最有利于提高系统吞吐量的作业调度算法是( **短作业优先算法** ), 能对紧急作业进行及时处理的调度算法是( **优先权高者优先算法** )。
- 9、在高响应比优先的调度算法中, 当各个作业的等待时间相同时, ( **短作业** )将得到优先调度, 当各个作业要求的运行时间相同时, ( **最先到达者(或等待时间最长的)** )将得到优先调度。

$$\text{优先权} = \frac{\text{等待时间} + \text{要求服务时间}}{\text{要求服务时间}}$$

1、设有三道作业, 它们的提交时间和运行时间如下表:

| 作业号 | 提交时刻<br>(时) | 运行时间(小时) |
|-----|-------------|----------|
| 1   | 10. 00      | 2        |
| 2   | 10. 10      | 1        |
| 3   | 10. 25      | 0. 25    |

求: 试给出下面两种调度算法下, 作业的执行顺序, 平均周转时间和带权平均周转时间。 (注意: 作业调度与进程调度均采用该调度算法, 内存无限大, 作业为纯计算型。要求写出每个作业的装入主存时间、开始执行时间, 结束执行时间, 周转时间和带权周转时间, 调度时间忽略不计, 保留小数点后两位)

(1) 先来先服务FCFS调度算法。

(2) 短作业优先SJF调度算法。

# (1) 先来先服务调度算法 (FCFS)

内存无限大, 作业调度和进程调度都采用FCFS

| 作业名  | 提交时间  | 运行时间 | 装入主存时间 | 开始执行时间 | 结束执行时间 | 周转时间 | 带权周转时间 |
|------|-------|------|--------|--------|--------|------|--------|
| 1    | 10.00 | 2    | 10.00  | 10.00  | 12.00  | 2    | 1      |
| 2    | 10.10 | 1    | 10.10  | 12.00  | 13.00  | 2.90 | 2.9    |
| 3    | 10.25 | 0.25 | 10.25  | 13.00  | 13.25  | 3    | 12     |
| 平均值: |       |      |        |        |        | 2.63 | 5.3    |

执行顺序: 1→2→3

周转时间=结束执行时间-提交时间

带权周转时间=周转时间/运行时间



## (2) 短作业优先调度算法 (SJF)

内存无限大, 作业调度和进程调度都采用SJF

| 作业名  | 提交时间  | 运行时间 | 装入主存时间 | 开始执行时间 | 结束执行时间 | 周转时间 | 带权周转时间 |
|------|-------|------|--------|--------|--------|------|--------|
| 1    | 10.00 | 2    | 10.00  | 10.00  | 12.00  | 2    | 1      |
| 2    | 10.10 | 1    | 10.10  | 12.25  | 13.25  | 3.15 | 3.15   |
| 3    | 10.25 | 0.25 | 10.25  | 12.00  | 12.25  | 2    | 8      |
| 平均值: |       |      |        |        |        | 2.38 | 4.05   |

执行顺序: 1→3→2

周转时间=结束执行时间-提交时间

带权周转时间=周转时间/运行时间

2、某一系统中有4个进程, 有3种资源 (R1, R2, R3), 进程的资源分配“瞬间状态”为:

|    | 最大需求<br>(Max) |   |   | 已分配<br>(Allocation) |   |   | 可用资源<br>(Available) |   |   |
|----|---------------|---|---|---------------------|---|---|---------------------|---|---|
| P1 | 3             | 2 | 2 | 1                   | 0 | 0 | 1                   | 1 | 2 |
| P2 | 6             | 1 | 3 | 5                   | 1 | 1 |                     |   |   |
| P3 | 3             | 1 | 4 | 2                   | 1 | 1 |                     |   |   |
| P4 | 4             | 2 | 2 | 0                   | 0 | 2 |                     |   |   |

使用银行家算法回答下列问题:

- (1) 计算数组还需要 (Need) ?
- (2) 系统此时是否安全?
- (3) 如果进程P2要求 (1, 0, 1), 系统能否立即满足进程的要求?  
为什么?
- (4) 如果进程P1要求 (1, 0, 1), 系统能否立即满足进程的要求?  
为什么? (不算(3))

(1) 需要 (Need)

|    | 最大需求<br>(Max) |   |   | 已分配<br>(Allocation) |   |   | 还需要<br>(Need) |   |   | 可用资源<br>(Available) |   |   |
|----|---------------|---|---|---------------------|---|---|---------------|---|---|---------------------|---|---|
| P1 | 3             | 2 | 2 | 1                   | 0 | 0 | 2             | 2 | 2 | 1                   | 1 | 2 |
| P2 | 6             | 1 | 3 | 5                   | 1 | 1 | 1             | 0 | 2 |                     |   |   |
| P3 | 3             | 1 | 4 | 2                   | 1 | 1 | 1             | 0 | 3 |                     |   |   |
| P4 | 4             | 2 | 2 | 0                   | 0 | 2 | 4             | 2 | 0 |                     |   |   |

(2) 利用安全算法对该时刻资源分配情况进行分析, 如下图所示:

|    | 最大需求<br>(Max) |   |   | 已分配<br>(Allocation) |   |   | 还需要<br>(Need) |          |          | 可用资源<br>(Available) |   |   |
|----|---------------|---|---|---------------------|---|---|---------------|----------|----------|---------------------|---|---|
| P1 | 3             | 2 | 2 | 1                   | 0 | 0 | 2             | 2        | 2        | 1                   | 1 | 2 |
| P2 | 6             | 1 | 3 | 5                   | 1 | 1 | <u>1</u>      | <u>0</u> | <u>2</u> |                     |   |   |
| P3 | 3             | 1 | 4 | 2                   | 1 | 1 | 1             | 0        | 3        |                     |   |   |
| P4 | 4             | 2 | 2 | 0                   | 0 | 2 | 4             | 2        | 0        |                     |   |   |

Work=Available=(1,1,2)      Finish[]

分配给P2,完成后Work=(6,2,3)      ture

分配给P3,完成后Work=(8,3,4)      ture

分配给P4,完成后Work=(8,3,6)      ture

分配给P1,完成后Work=(9,3,6)      ture

由以上分析可知, 在该时刻存在着一个安全序列  
{P2, P3, P4, P1}, 故系统是安全的。

(3)  $P_2$ 请求资源: $P_2$ 发出请求向量 $Request_2(1, 0, 1)$ , 系统按银行家算法进行检查:

①  $Request_2(1, 0, 1) \leq Need_2(1, 0, 2)$

②  $Request_2(1, 0, 1) \leq Available(1, 1, 2)$

③ 系统先假定可为 $P_2$ 分配资源, 并修改 $Available$ ,  $Allocation_2$ 和 $Need_2$ 向量, 由此形成的资源变化情况:

|    | 最大需求 |   |   | 已分配 |   |   | 还需求 |   |   | 可用资源向量 |   |   |
|----|------|---|---|-----|---|---|-----|---|---|--------|---|---|
| P2 | 6    | 1 | 3 | 6   | 1 | 2 | 0   | 0 | 1 | 0      | 1 | 1 |

④ 再利用安全性算法检查此时系统是否安全。

Request<sub>2</sub>(1,0,1)

|    | 最大需求<br>(Max) |   |   | 已分配<br>(Allocation) |   |   | 还需要<br>(Need) |   |   | 可用资源<br>(Available) |   |   |
|----|---------------|---|---|---------------------|---|---|---------------|---|---|---------------------|---|---|
| P1 | 3             | 2 | 2 | 1                   | 0 | 0 | 2             | 2 | 2 | 0                   | 1 | 1 |
| P2 | 6             | 1 | 3 | 6                   | 1 | 2 | 0             | 0 | 1 |                     |   |   |
| P3 | 3             | 1 | 4 | 2                   | 1 | 1 | 1             | 0 | 3 |                     |   |   |
| P4 | 4             | 2 | 2 | 0                   | 0 | 2 | 4             | 2 | 0 |                     |   |   |

Work=Available=(0,1,1)      Finish[]

分配给P2,完成后Work=(6,2,3)      ture

分配给P3,完成后Work=(8,3,4)      ture

分配给P4,完成后Work=(8,3,6)      ture

分配给P1,完成后Work=(9,3,6)      ture

由以上分析可知, 在该时刻存在着一个安全序列  
{P2, P3, P4, P1}, 故系统是安全的, 可以分配。

(4)  $P_1$ 请求资源: $P_1$ 发出请求向量 $Request_1(1, 0, 1)$ , 系统按银行家算法进行检查:

①  $Request_1(1, 0, 1) \leq Need_1(2, 2, 2)$

②  $Request_1(1, 0, 1) \leq Available(1, 1, 2)$

③系统先假定可为 $P_1$ 分配资源, 并修改 $Available$ ,  $Allocation_1$ 和 $Need_1$ 向量, 由此形成的资源变化情况:

|    | 最大需求  | 已分配   | 还需求   | 可用资源向量 |
|----|-------|-------|-------|--------|
| P1 | 3 2 2 | 2 0 1 | 1 2 1 | 0 1 1  |

④再利用安全性算法检查此时系统是否安全。

|    | 最大需求<br>(Max) |   |   | 已分配<br>(Allocation) |   |   | 还需要<br>(Need) |   |   | 可用资源<br>(Available) |   |   |
|----|---------------|---|---|---------------------|---|---|---------------|---|---|---------------------|---|---|
| P1 | 3             | 2 | 2 | 2                   | 0 | 1 | 1             | 2 | 1 | 0                   | 1 | 1 |
| P2 | 6             | 1 | 3 | 5                   | 1 | 1 | 1             | 0 | 2 |                     |   |   |
| P3 | 3             | 1 | 4 | 2                   | 1 | 1 | 1             | 0 | 3 |                     |   |   |
| P4 | 4             | 2 | 2 | 0                   | 0 | 2 | 4             | 2 | 0 |                     |   |   |

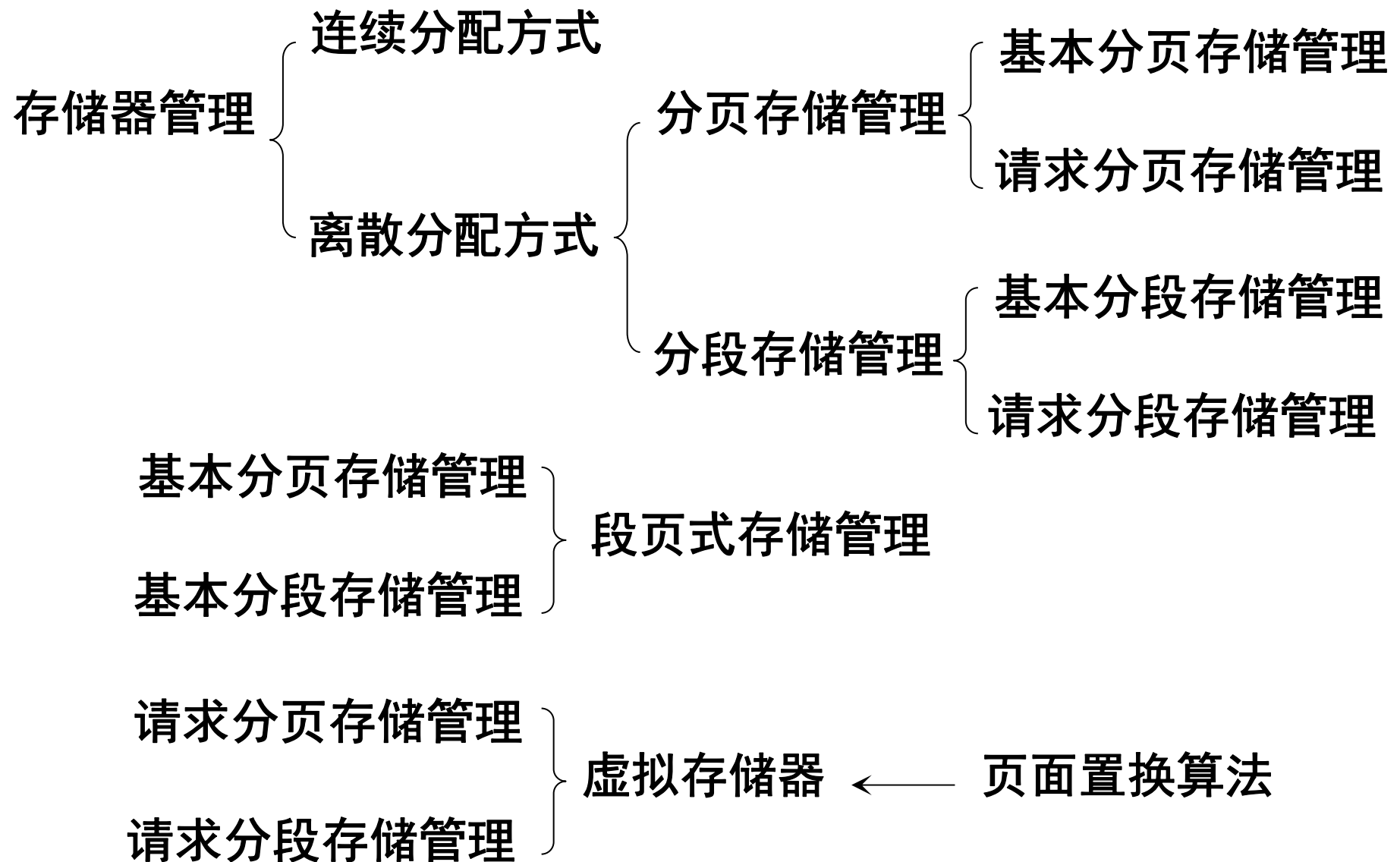
Work=Available=(0,1,1)      Finish[]

不能满足任何进程的需要,故系统进入不安全状态,此时系统不能分配资源给P1



# 第四章存储器管理——小结

- 存储器管理的**主要任务**是分配存储器
- **主要目的**是提高存储器的利用率。
- 它的主要功能有：存储器的分配与回收、地址转换与保护、主存的扩充。
- 熟悉和掌握以下基本概念：
  - 逻辑地址、物理地址、地址转换、静态重定位、动态重定位、碎片、对换、连续分配、离散分配、虚拟存储器、请求分页存储管理、请求分段存储管理等
- 熟悉和掌握以下基本知识：
  1. 连续存储管理方式
  2. 离散存储管理方式
  3. 虚拟存储管理方式



1、采用连续分配方式的动态分区分配, 对如图所示的内存分配情况(其中, 阴影部分表示已占用分区, 空白部分表示空闲分区), 若要申请一块40K的内存, 对于最佳适应算法给出分配区域的首地址是( C )

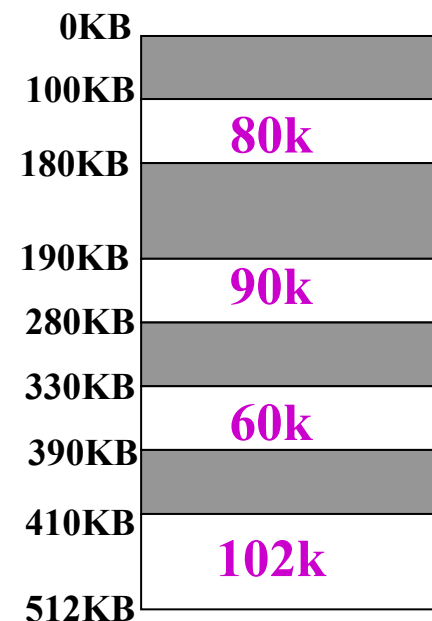
A. 100KB      B. 190KB      C. 330KB      D. 410KB

2、采用连续分配方式的动态分区分配, 在右图所示中, 若要申请一块40KB的内存, 使首地址最大的分配算法( C )

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

3、下列算法中最有可能会产生“抖动”现象的是( A )。

A. 先进先出页面置换算法  
B. 最近最久未使用置换算法  
C. 最少使用页面置换算法  
D. 最佳页面置换算法



4、存储管理是对内存中( **B** )区域进行管理。

- A. 整个内存
- B. 供用户使用的
- C. 供系统使用的
- D. 供程序使用的

5、下面是关于存储管理功能的论述, 正确的论述是( **A** )和( **D** )

- A. 内存分配最基本的任务是为每道程序分配内存空间, 其它追求的主要目标是提高存储空间的利用率。
- B. 为了提高内存保护的灵活性, 内存保护通常由软件实现。
- C. 对换技术已不是现代操作系统中常用的一种技术。
- D. 地址映射是指将程序空间中的逻辑地址变为内存空间的物理地址。
- E. 虚拟存储器是在物理上扩充内存容量。

6、在下列存储管理方案中, 不适用于多道程序的是( **A** )

- A. 单一连续分配
- B. 固定分区分配
- C. 可重定位分区分配
- D. 段页式存储分配

7、在固定分区分配中, 每个分区的大小是( C )

- A. 相同的
- B. 可以不同但作业长度固定
- C. 可以不同但预先固定
- D. 根据用户要求而定

8、在固定分区分配中, 为了提高内存的利用率, 可采用如下技术( A )

- A. 按经常出现的作业大小来划分分区
- B. 划分分区都相同
- C. 不同请求队列中的作业可以申请相同的分区
- D. 大作业可以申请多个分区

9、采用固定分区分配的最大缺点是( C )

- A. 不利于内存的保护
- B. 分配算法复杂
- C. 内存的利用率不高
- D. 零头太多

10、可重定位分区分配采用的地址转换公式是( C )

- A. 绝对地址=上界寄存器值+逻辑地址
- B. 绝对地址=下界寄存器值+逻辑地址
- C. 绝对地址=重定位寄存器值+逻辑地址
- D. 绝对地址=块号\*块长+页内地址

11、对外存对换区的管理以( **D** )为主要目标, 对外存文件区的管理以( **B** )为主要目标。

- A. 提高系统吞吐量
- B. 提高存储空间的利用率
- C. 降低存储费用
- D. 提高换入/换出的速度

12、采用分页存储管理使处理器执行指令的速度( **B** )

- A. 提高
- B. 降低
- C. 不定
- D. 不受影响

13、分段存储管理中, 分段是由用户决定的, 因此( **B** )

- A. 段内的地址和段间的地址都是连续的
- B. 段内的地址是连续的, 而段间的地址是不连续的
- C. 段内的地址是不连续的, 而段间的地址是连续的
- D. 段内的地址和段间的地址都不是连续的

14、在请求分页管理中, 在页表中增加了若干项, 其中状态位供( **C** )时参考, 修改位供( **B** )和( **D** )时参考, 访问字段供( **B** )和( **D** )时参考, 外存地址供( **E** )时参考。

- A. 分配页面
- B. 置换算法
- C. 程序访问
- D. 换出页面
- E. 调入页面

15、请求分页管理中, 缺页中断率与进程所得的内存物理块数, ( **B** ) 和 ( **C** ) 等因素有关。

A. 页表的位置    B. 置换算法    C. 页面大小    D. 进程调度算法

16、请求分页管理中, 页面的大小与可能产生的缺页中断次数 ( **B** )

A. 成正比    B. 成反比    C. 无关    D. 成固定比值

17、下列说法正确的是 ( **B** )

A. 在段页式系统中, 以页为单位管理用户的虚拟空间, 以段为单位管理内存空间。

B. 在段页式系统中, 以段为单位管理用户的虚拟空间, 以页为单位管理内存空间。

C. 为提高请求分页系统中内存的利用率, 允许用户使用不同大小的页面。

D. 在虚拟存储器中, 为了能让更多的作业同时运行, 通常只装入10%的作业后便运行。

18、在下面的存储管理方案中, 可以使用上下界地址寄存器实现存储保护的是( **A** )和( **B** )

- A. 固定分区存储管理
- B. 动态分区存储管理
- C. 分页存储管理
- D. 分段存储管理

19、虚拟存储器最基本的特征是((1) **B** ), 该特征主要是基于((2) **E** ), 实现虚拟存储器最关键的技术是((3) **C** )。

(1) A. 一次性 B. 多次性 C. 离散性 D. 驻留性

(2) A. 计算机的高速性 B. 大容量内存 C. 大容量硬盘  
D. 循环性原理 E. 局部性原理

(3) A. 内存分配 B. 置换算法 C. 请求调页(段) D. 对换空间管理

20、碎片最严重的存储管理方式是( **A** )

- A. 固定分区
- B. 动态分区
- C. 分页管理
- D. 分段



- 21、在动态分区分配的算法中,首次适应算法倾向于优先利用内存中的( 低地址 )部分的空闲分区,从而保留了( 高地址 )部分的空闲分区。
- 22、地址变换机构的最基本任务是将(用户地址空间)中的( 逻辑地址(或相对地址) )变换为( 内存空间 )中的( 物理地址(或绝对地址) )。在分页系统中为实现地址变换而设置了页表寄存器,其中存放了( 页表始址 )和( 页表长度 ),在进程未运行时,他们存放在( PCB(或进程控制块) )中。在分页系统中进行地址变换时,应将页表寄存器中的( 页表始址 )和( 页号\*页表项长度 )进行相加,得到该页的在页表中的位置,从中可得到( 物理块号 )。
- 23、为实现请求分页管理,应在页表中增加( 状态位 )、( 访问字段 )、( 修改位 )、( 外存地址 )几项。

24、假定某分页存储管理的系统中, 内存容量为1MB, 被分成256块, 块号为0, 1, 2, . . . . ., 255。某作业的地址空间占4页, 其页号为0, 1, 2, 3, 被分配到内存的2, 4, 1, 5块中。则有:

(1) 内存地址应该用( **20** )位来表示。

(2) 作业每一页的长度为( **4KB** ), 逻辑地址中的页内地址应用( **12** )位表示。

(3) 把作业中每一页在分到的内存中的起始地址填入下表:

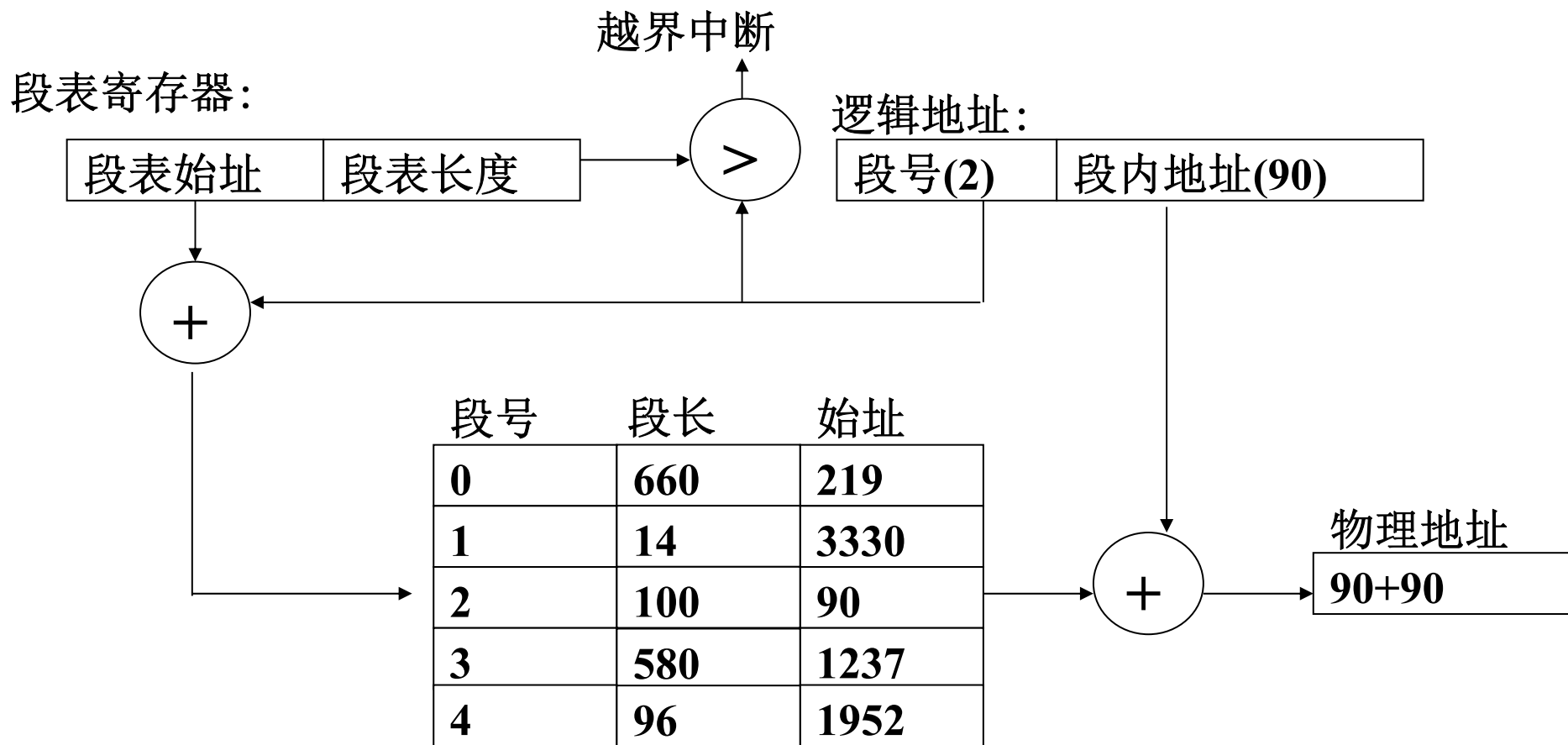
| 页号       | 起始地址         |
|----------|--------------|
| <b>0</b> | <b>8 KB</b>  |
| <b>1</b> | <b>16 KB</b> |
| <b>2</b> | <b>4 KB</b>  |
| <b>3</b> | <b>20 KB</b> |

25、某分段存储管理中采用如下表所示的段表：

- (1) 给定段号2和段内地址90(单位字节), 说明分段管理中的地址变换过程(没有快表的)?
- (2) 计算[0, 430], [1, 10], [2, 500], [3, 400], [4, 20], [5, 100], 的内存地址, 其中方括号内的第一元素是段号, 第二元素是段内地址(单位字节)。
- (3) 说明存取主存中的一条指令或数据至少要访问几次主存。

| 段号 | 段的长度<br>(字节) | 内存起始地址<br>(字节) |
|----|--------------|----------------|
| 0  | 660          | 219            |
| 1  | 14           | 3330           |
| 2  | 100          | 90             |
| 3  | 580          | 1237           |
| 4  | 96           | 1952           |

答：(1) 地址变换图：



| 段号 | 段的长度<br>(字节) | 内存起始地址<br>(字节) |
|----|--------------|----------------|
| 0  | 660          | 219            |
| 1  | 14           | 3330           |
| 2  | 100          | 90             |
| 3  | 580          | 1237           |
| 4  | 96           | 1952           |

(2) [0, 430]——>物理地址:  $219+430=649$

[1, 10]——>物理地址:  $3330+10=3340$

[2, 500]——>物理地址: 由于 $500>100$ ,所以段内地址越界

[3, 400]——>物理地址:  $1237+400=1637$

[4, 20]——>物理地址:  $1952+20=1972$

[5, 100]——>物理地址: 由于 $5>4$ ,所以段号越界

(3) 两次: 一次是访问段表, 另一次是访问需要的指令或数据。

26、若在一分页存储管理系统中,某作业的页表如下所示。  
已知页面大小为1024字节,试将逻辑地址  
[0, 100], [1, 179], [2, 785]和[3, 1010], 页内地址单位字  
节, 转化为相应的物理地址。

| 页号 | 块号 |
|----|----|
| 0  | 3  |
| 1  | 5  |
| 2  | 6  |
| 3  | 2  |

答:

- (1) 对于逻辑地址 [0, 100], 查页表第0页在第3块, 所以物理地址  
为:  $1024 \times 3 + 100 = 3172$
- (2) 对于逻辑地址 [1, 179], 查页表第1页在第5块, 所以物理地址  
为:  $1024 \times 5 + 179 = 5299$
- (3) 对于逻辑地址 [2, 785], 查页表第2页在第6块, 所以物理地址  
为:  $1024 \times 6 + 785 = 6929$
- (4) 对于逻辑地址 [3, 1010], 查页表第3页在第2块, 所以物理地  
址为:  $1024 \times 2 + 1010 = 3058$

27、考虑一个由8个页面, 每页有1024个字节组成的逻辑空间, 把它装入到有32个物理块的存储器中, 问:

(1) 逻辑地址需要多少位表示? (二进制)

(2) 绝对地址需要多少位表示? (二进制)

答:

因为页面数为 $8=2^3$ , 故需要3位二进制数表示。每页有1024个字节,  $1024=2^{10}$ , 于是页内地址需要10位二进制数表示。32个物理块, 需要5位二进制数表示 ( $32=2^5$ )。

(1) 页的逻辑地址由页号和页内地址组成, 所以需要 $3+10=13$ 位二进制数表示。

(2) 页的绝对地址由块号和页内地址的拼接, 所以需要 $5+10=15$ 位二进制数表示。

28、某虚拟存储器的用户编程空间共32个页面, 每页为1KB, 内存为16KB。假定某时刻一用户页表中已调入内存的页面的页号和物理块号的对照表如下:

| 页号 | 物理块号 |
|----|------|
| 0  | 5    |
| 1  | 10   |
| 2  | 4    |
| 3  | 7    |

问:

- (1) 逻辑地址的有效位是( **15** )位?
- (2) 物理地址需要( **14** )位?
- (3) 则逻辑地址0A5CH所对应的物理地址是什么? (要求写出计算步骤)
- (4) 则逻辑地址093CH所对应的物理地址是什么? (要求写出计算步骤)



解:[分析]分页存储管理的逻辑地址分为两部分:页号和页内地址。

由已知条件“用户编程空间共32个页面”,可知页号部分占5位;由“每页为1KB”, $1K=2^{10}$ ,可知页内地址占10位。由“内存为16KB”,可知有16块,块号为4位。

(3) 逻辑地址0A5CH所对应的二进制表示形式是:

0000 1010 0101 1100, 根据上面的分析, 下划线部分为页内地址, 编码“0000 10”为页号, 表示该逻辑地址对应的页号为2。查页表, 得到物理块号是4(十进制), 即物理块地址为:01 00, 拼接块内地址10 0101 1100, 得01 0010 0101 1100, 即125CH。

(4) 逻辑地址093CH所对应的二进制表示形式是:

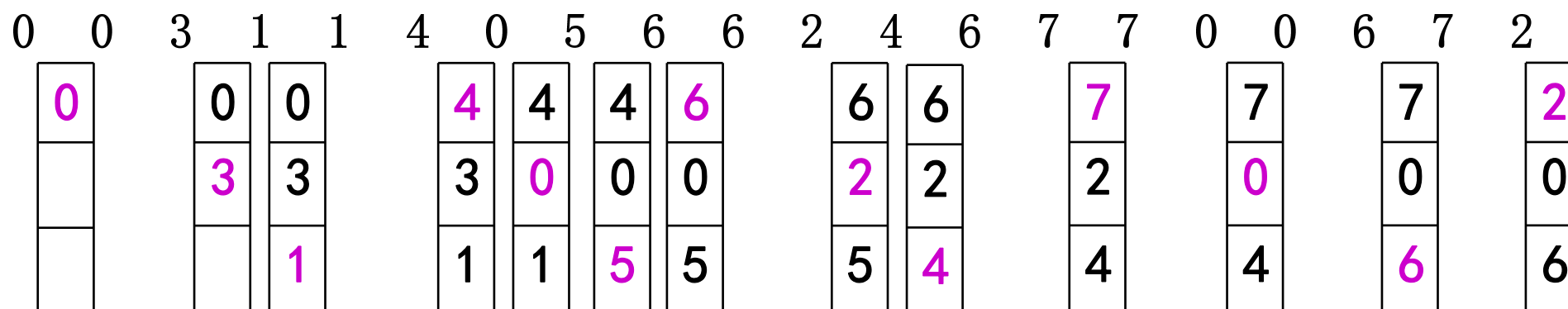
0000 1001 0011 1100, 根据上面的分析, 下划线部分为页内地址, 编码“0000 10”为页号, 表示该逻辑地址对应的页号为2。查页表, 得到物理块号是4(十进制), 即物理块地址为:01 00, 拼接块内地址01 0011 1100, 得01 0001 0011 1100, 即113CH。

29、已知某进程访问以下页面:0、0、3、1、1、4、0、5、6、6、2、4、6、7、7、0、0、6、7、2, 如果进程有3个页框可用且使用下列置换算法, 求出现缺页的次数及缺页率。

(1) 先进先出FIFO置换算法

(2) 最近最久未使用LRU置换算法

## (1)先进先出FIFO置换算法



缺页次数: 13次 (算前3个), 缺页率:  $13/20=65\%$

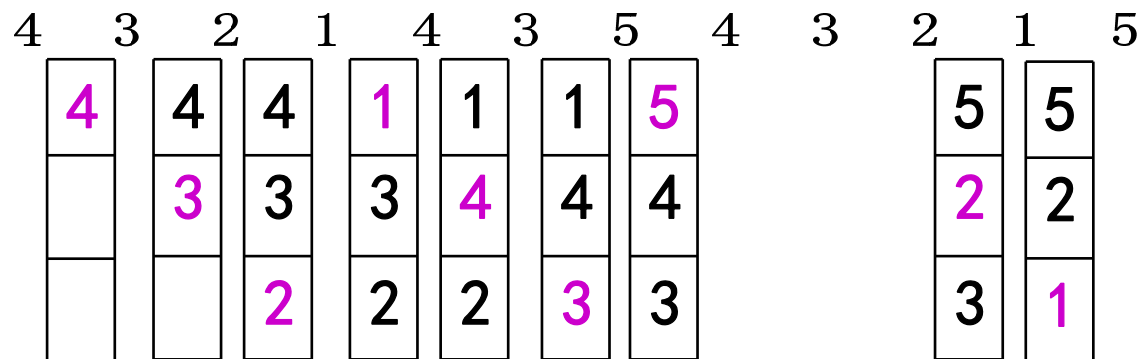
## (2)最近最久未使用LRU置换算法

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 3 | 1 | 1 | 4 | 0 | 5 | 6 | 6 | 2 | 4 | 6 | 7 | 7 | 0 | 0 | 6 | 7 | 2 |
| 0 |   | 0 | 0 |   | 4 | 4 | 4 | 6 |   | 6 | 6 |   | 6 |   | 6 |   |   |   | 6 |
|   |   | 3 | 3 |   | 3 | 0 | 0 | 0 |   | 2 | 2 |   | 7 |   | 7 |   |   |   | 7 |
|   |   |   | 1 |   | 1 | 1 | 5 | 5 |   | 5 | 4 |   | 4 |   | 0 |   |   |   | 2 |

缺页次数:12次(算前3个), 缺页率:12/20=60%

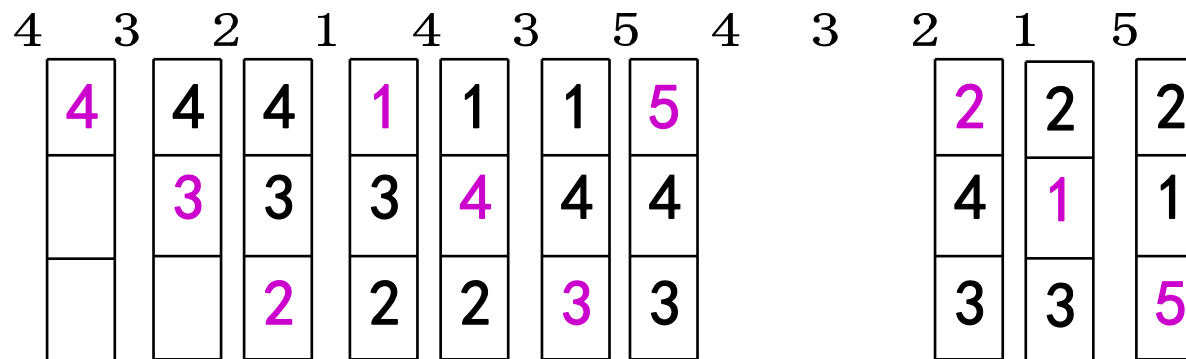
30、已知某进程访问以下页面:4、3、2、1、4、3、5、4、3、2、1、5, 当分配该进程的物理块数M分别是3和4时, 分别采用先进先出FIFO置换算法和最近最久未使用LRU置换算法, 求出现缺页的次数及缺页率, 并比较所得结果。

# (1)先进先出FIFO置换算法(3个物理块时)



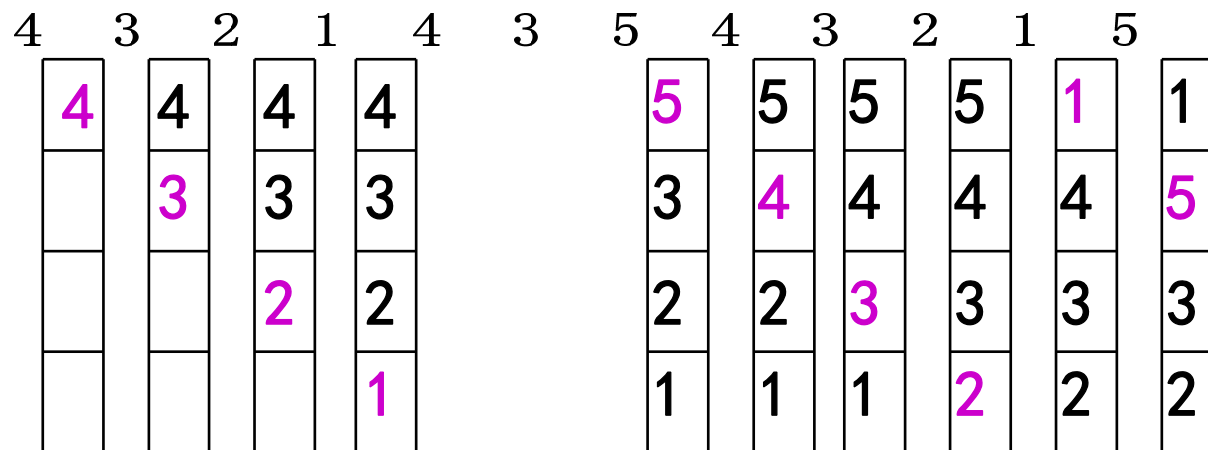
缺页次数:9次(算前3个), 缺页率:9/12=75%

## (2)最近最久未使用LRU置换算法(3个物理块时)



缺页次数: 10次 (算前3个), 缺页率:  $10/12=83\%$

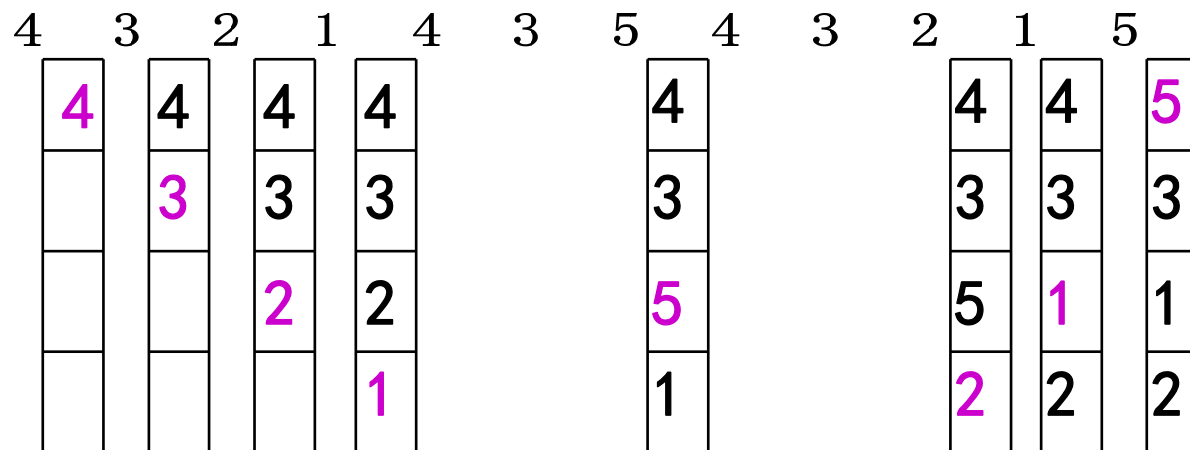
### (3)先进先出FIFO置换算法(4个物理块时)



缺页次数: 10次 (算前4个), 缺页率:  $10/12=83\%$



#### (4)最近最久未使用LRU置换算法(4个物理块时)



缺页次数: 8次 (算前4个), 缺页率:  $8/12=67\%$

通过以上缺页次数和缺页率的分析计算,可以看出,对于最近最久未使用LRU置换算法,增加物理块数,可以减少缺页次数,降低缺页率,而对先进先出FIFO置换算法,增加物理块数,不一定能减少缺页次数。

# 第五章设备管理——小结

- 设备管理的**主要任务**是分配I/O设备。
- 主要**目的**是提高I/O设备的使用效率。
- 它的**主要功能**有:缓冲区管理、设备分配、设备处理、虚拟设备及实现设备独立性。
- I/O4种控制方式:**程序I/O方式、中断驱动I/O方式、直接存储访问DMA I/O控制方式和I/O通道控制方式**
- 缓冲技术的概念及4种常见的缓冲技术:单缓冲、双缓冲、循环缓冲及缓冲池
- 设备分配、设备处理、SPooling技术
- 磁盘调度算法

1、通过硬件和软件的功能扩充,把原来独占的设备改造成若干用户共享的设备,这种设备称为( C )

A. 存储设备 B. 系统设备 C. 虚拟设备 D. 用户设备

2、CPU输出数据的速度远远高于打印机的打印速度,为解决这一矛盾,可采用( C )

A. 并行技术 B. 通道技术 C. 缓冲技术 D. 虚拟存储技术

3、为了使多个进程能有效地同时处理I/O,最好使用( A )结构的缓冲技术。

A. 缓冲池 B. 单缓冲区 C. 双缓冲区 D. 循环缓冲区

4、磁盘属于((1) **C** ),信息的存取是以((2) **D** )单位进行的,磁盘的I/O控制主要采取((3) **C** )方式,打印机的I/O控制主要采取((3) **B** )方式。

(1) A. 字符设备      B. 独占设备      C. 块设备      D. 虚拟设备

(2) A. 位      B. 字节      C. 帧      D. 数据块

(3) A. 程序I/O方式      B. 中断驱动I/O方式      C. DMA      D. SP00Ling

5、下面关于设备属性的论述中正确的为( **B** )

A. 字符设备的一个基本特征是可寻址的,即能指定输入时的源地址和输出时的目标地址

B. 共享设备必须是可寻址的和可随机访问的设备

C. 共享设备是指在同一时刻内,允许多个进程同时访问的设备

D. 在分配共享设备和独占设备时,都可能引起进程死锁

6、下面关于虚拟设备的论述中, 正确的是( B )

- A. 虚拟设备是指允许用户使用比系统中具有的物理设备更多的设备
- B. 虚拟设备是指把一个物理设备变成多个对应的逻辑设备
- C. 虚拟设备是指允许用户以标准化方式来使用物理设备
- D. 虚拟设备是指允许用户程序不必全部装入内存便可使用系统中的设备

7、通道是一种特殊((1) C ), 具有((2) A )能力, 它用于实现((3) A )之间的信息传输。

(1) A. I/O设备    B. 设备控制器    C. 处理机    D. I/O控制器

(2) A. 执行I/O指令集    B. 执行CPU指令集  
C. 传输I/O指令    D. 运行I/O进程

(3) A. 内存与外设    B. CPU与外设    C. 内存与外存    D. CPU与外存

8、为实现设备分配, 应为每一个设备设置一张((1) A ), 在系统中配置一张((1) C ), 为实现设备的独立性, 系统中应设置一张((2) D )

- |              |           |
|--------------|-----------|
| (1) A. 设备控制表 | B. 控制器控制表 |
| C. 系统设备表     | D. 设备分配表  |
| (2) A. 设备开关表 | B. I/O请求表 |
| C. 系统设备表     | D. 逻辑设备表  |

9、下面不适合磁盘调度算法的是( D )

- |         |         |         |            |
|---------|---------|---------|------------|
| A. FCFS | B. SCAN | C. SSTF | D. 时间片轮转算法 |
|---------|---------|---------|------------|

10、在大型系统中, 主机通常是通过( 通道 )和( 设备控制器 )与外设相连。

- 11、操作系统中通常采用的I/O控制方式有(程序I/O方式)、(中断驱动I/O控制方式)、(直接存储器访问DMA I/O控制方式)和(I/O通道控制方式)。在对打印机进行I/O控制时,通常采用(中断驱动I/O控制方式),对磁盘的I/O控制采用(DMA)控制方式。而DMA是指允许(外部设备)与(内存)之间直接交换数据的控制器,在DMA中必须设置地址寄存器,用于存放(内存起始地址)。
- 12、SPooling系统是由磁盘中的(输入井)和(输出井),内存中的(输入缓冲区)和(输出缓冲区),以及(输入进程)和(输出进程)所构成。SPooling是对脱机I/O工作方式的模拟,SPooling系统中的(输入井)和(输出井)是对脱机输入输出中的磁盘进行模拟,(输入进程)和(输出进程)是对脱机输入输出中的外围控制机进行模拟。



13、设备管理中引入缓冲区机制的主要原因是

( 缓和CPU与I/O设备之间速度不匹配矛盾 )、  
( 减少对CPU的中断频率, 放宽对CPU中断响应时间的限制 )  
和( 提高CPU和I/O设备之间的并行性 )。

14、设备管理的主要功能是(缓冲区管理 )、( 设备分配 )、  
( 设备处理)、(虚拟设备)及( 实现设备独立性 )等。

15、假定一磁盘有200个柱面, 编号为0—199, 在完成了磁道125处的请求后, 当前正在磁道143处为一个请求服务。若请求队列的先后顺序为

86, 147, 91, 177, 94, 150, 102, 175, 130

试分别采用FCFS(先来先服务)、SSTF(最短寻道时间优先)、SCAN(扫描)算法和CSCAN(循环扫描)完成上述请求, 写出磁头移动的顺序, 并计算存取臂移动总量(单位为磁道数)。

答:采用FCFS算法调度时(当前143)

被访问的下一个磁道号

移动距离(磁道数)

86

57

147

61

91

56

177

86

94

83

150

56

102

48

175

73

130

45

---

总移动量:565

采用SSTF算法调度时(当前143)

被访问的下一个磁道号

移动距离(磁道数)

147

4

150

3

130

20

102

28

94

8

91

3

86

5

175

89

177

2

---

总移动量:162

采用SCAN算法调度时(当前143)

被访问的下一个磁道号

移动距离(磁道数)

147

4

150

3

175

25

177

2

130

47

102

28

94

8

91

3

86

5

---

总移动量:125

采用CSCAN算法调度时(当前143)

被访问的下一个磁道号

移动距离(磁道数)

147

4

150

3

175

25

177

2

86

91

91

5

94

3

102

8

130

28

---

总移动量:169

# 第六章文件管理——小结

- 文件管理的**主要任务**是分配外存空间, 对用户文件和系统文件进行管理, 方便用户使用, 并保证文件的安全性。
- 主要**目的**是提高外存的使用效率和方便用户读文件的使用。
- 主要**功能**是文件存储空间的管理、目录的管理、文件读/写的管理、文件共享和保护。

熟悉和掌握内容:

- 文件、记录、数据项、文件分类、文件的组织(逻辑和物理结构)
- 文件存储空间的管理
- 熟悉文件目录的管理(单级、两级、多级目录结构)、文件的共享与保护

1、在下列文件的外存分配方式中,不利于文件长度动态增长的文件物理结构是( A )。

A. 连续分配 B. 链接分配 C. 索引分配 D. 以上都不对

2、文件系统中若文件的外存分配方式采用连续分配,则文件控制块FCB中有关文件的物理位置的信息应包括( B )。

(I)起始块号                      (II)文件长度                      (III)索引表地址

A. 全部 B. (I)和(II) C. (I)和(III) D. (II)和(III)

3、操作系统为保证未经文件拥有者授权,任何其他用户不能使用该文件所提供的解决方法是( A )。

A. 文件保护 B. 文件保密 C. 文件转储 D. 文件共享



4、文件系统最基本的目标是((1) **A** ),它主要是通过  
( (2) **B** )功能实现的,文件系统所追求的最重要目标是  
( (1) **D** )。

- (1)     A. 按名存取                      B. 文件共享  
          C. 文件保护                    D. 提高对文件的存取速度
- (2)     A. 存储空间管理                B. 目录管理  
          C. 文件读写管理                D. 文件安全管理

5、按逻辑结构可把文件分为( **E** )和( **F** )两类。

- A. 读、写文件                      B. 只读文件                      C. 索引文件
- D. 链式文件                        E. 记录式文件                    F. 流式文件

- 6、下面关于顺序文件和链接文件的论述中正确的是( C )。
- A. 顺序文件只能建立在顺序存储设备上, 而不能建立在磁盘上。
  - B. 在显示链接文件中是在每个盘块中设置一链接指针, 用于将文件的所有盘块链接起来。
  - C. 顺序文件采用连续分配方式, 而链接文件和索引文件则都可采用离散分配方式。
  - D. 在MS-DOS中采用的是隐式链接文件结构。
- 7、下面关于索引文件的论述中正确的是( B )。
- A. 在索引文件中, 索引表的每个表项中必须含有相应记录的关键字和存放该记录的物理地址。
  - B. 对顺序文件进行检索时, 首先从FCB中读出文件的第一个盘块号, 而对索引文件进行检索时, 应先从FCB中读出文件索引表始址。
  - C. 对于一个具有三级索引表的文件, 存取一个记录必须要访问三次磁盘。
  - D. 在文件较大时, 进行顺序存取比随机存取快。

8、在存取文件时,如果利用给定的记录值对链表或索引表进行检索,以找到指定记录的物理地址,则上述文件分别称为( B )或( C ),如果根据给定的记录键值直接获得指定记录的物理地址,则把这种文件称为( D )。

A. 顺序文件 B. 链接文件 C. 索引文件 D. 直接文件

9、在文件管理中,位示图主要是用于( B )。

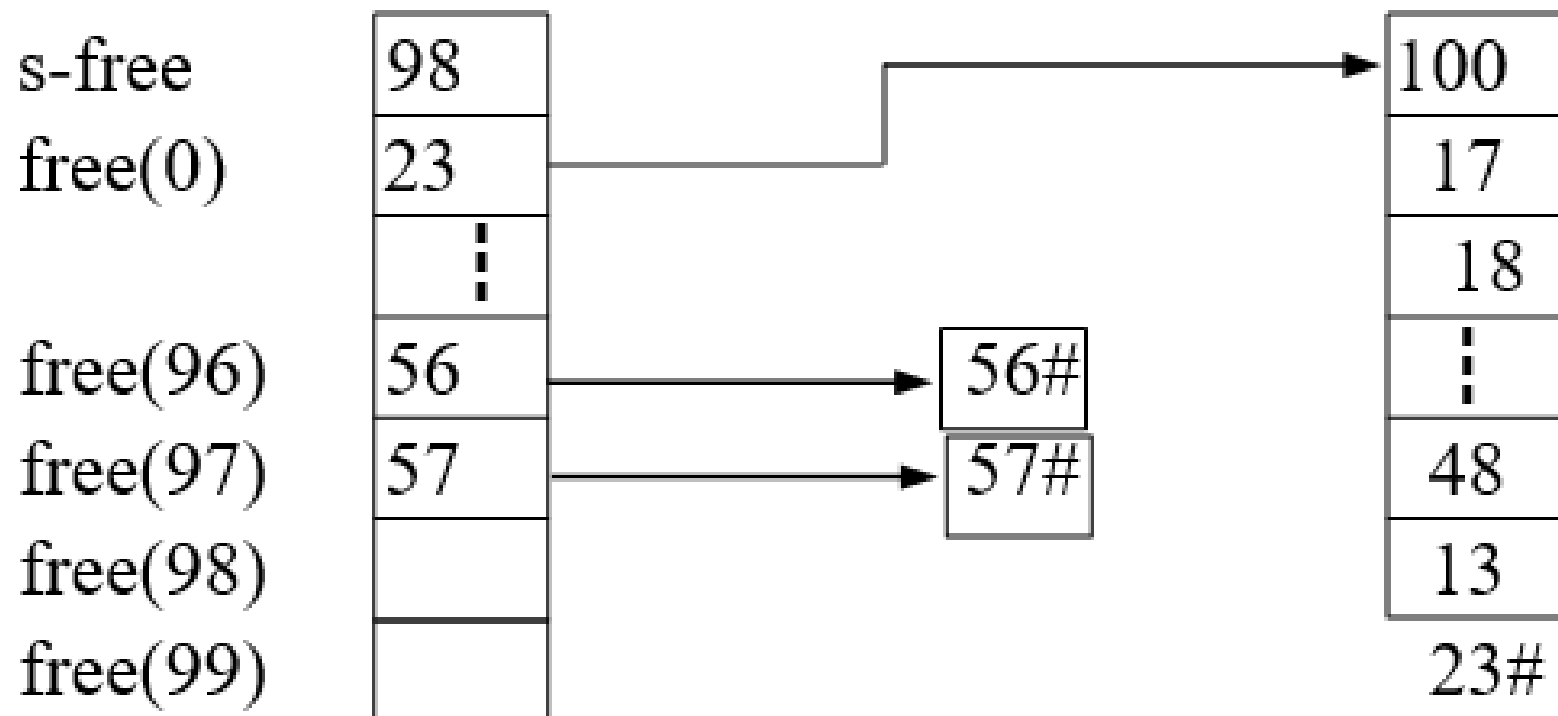
A. 磁盘的驱动调动 B. 磁盘空间的分配和回收  
C. 文件目录的查找 D. 页面置换

10、用( B )可以防止共享文件可能造成的破坏,但实现起来系统开销太大。

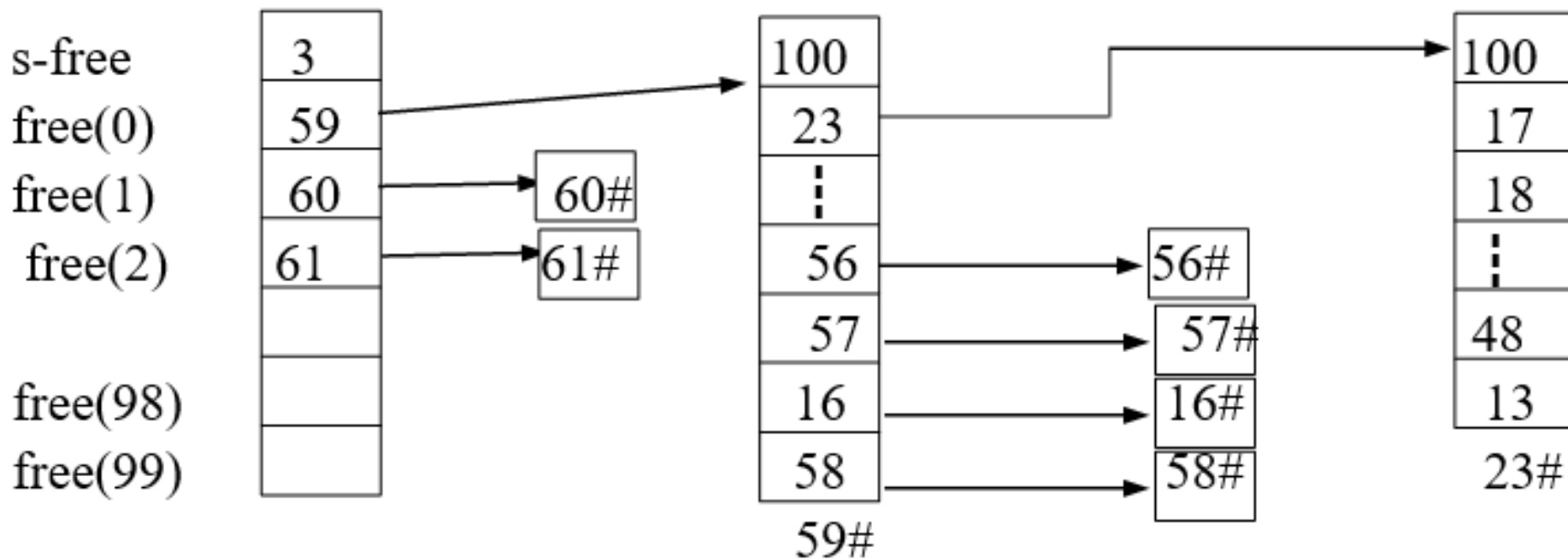
A. 用户对数型目录结构中目录和文件的许可权规定  
B. 存取控制表  
C. 定义不同用户对文件的使用权  
D. 隐蔽文件目录

- 11、在利用基本文件目录法实现文件共享时, 文件系统必须设置一个( 基本文件目录 ), 每个用户都应具有一个( 符号文件目录 )。
- 12、文件管理的基本功能有( 文件存储空间的管理 )、( 目录管理 )、( 文件的读/写管理 )和( 文件共享保护 )。
- 13、记录是一组相关( 数据项 )的集合。文件是具有( 文件名 )的一组相关( 元素 )的集合。
- 14、可将链接式文件中的各记录装入到( 离散的 )多个盘块中, 并通过( 链接指针 )将它们构成一个队列, ( 显示链接 )具有较高的检索速度。可将索引文件中的各记录装入到( 离散的 )多个盘中, 为每个文件建立一张( 索引表 )。

15、当UNIX文件系统的外存资源分配情况处于图示状态时，首先由进程M释放5个物理块16、58、59、60、61，再由进程N申请6个物理块，试画图说明回收和分配物理块时的状态。



进程M释放5个物理块后的状态。



进程N申请6个物理块后的状态。

