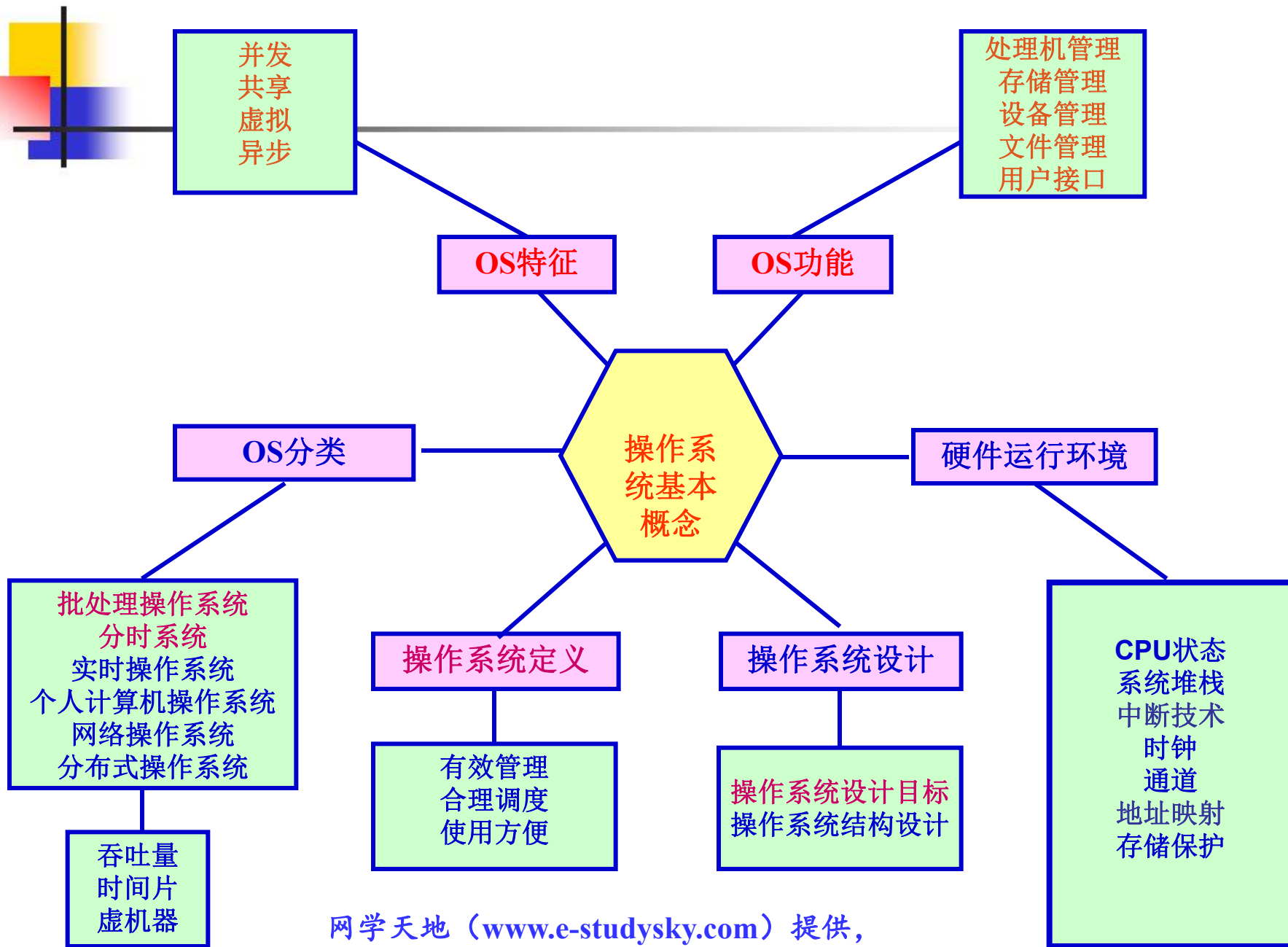
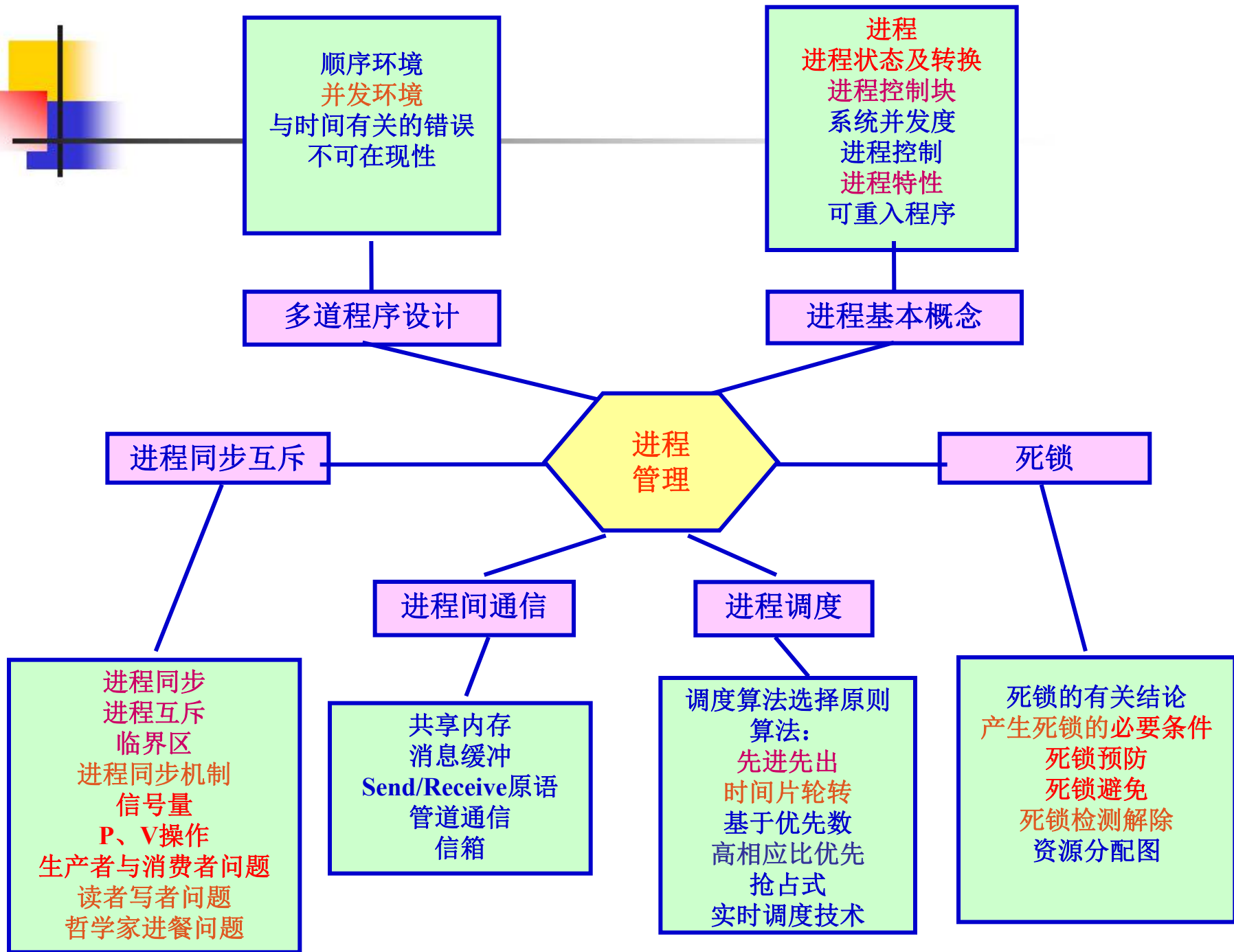
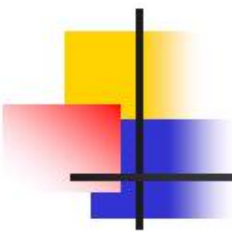
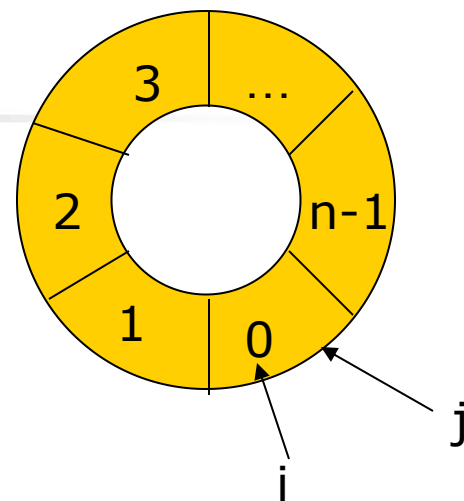


网学天地 (www.e-studysky.com) 提供,
咨询QQ: 2696670126





一、生产者—消费者问题



- 同步关系: $P \rightarrow C$
- 互斥关系: 互斥访问BUF

设: 生产进程资源私用量 e : BUF中空buf数
消费进程资源私用量 f : BUF中产品数目

$$e + f = n$$

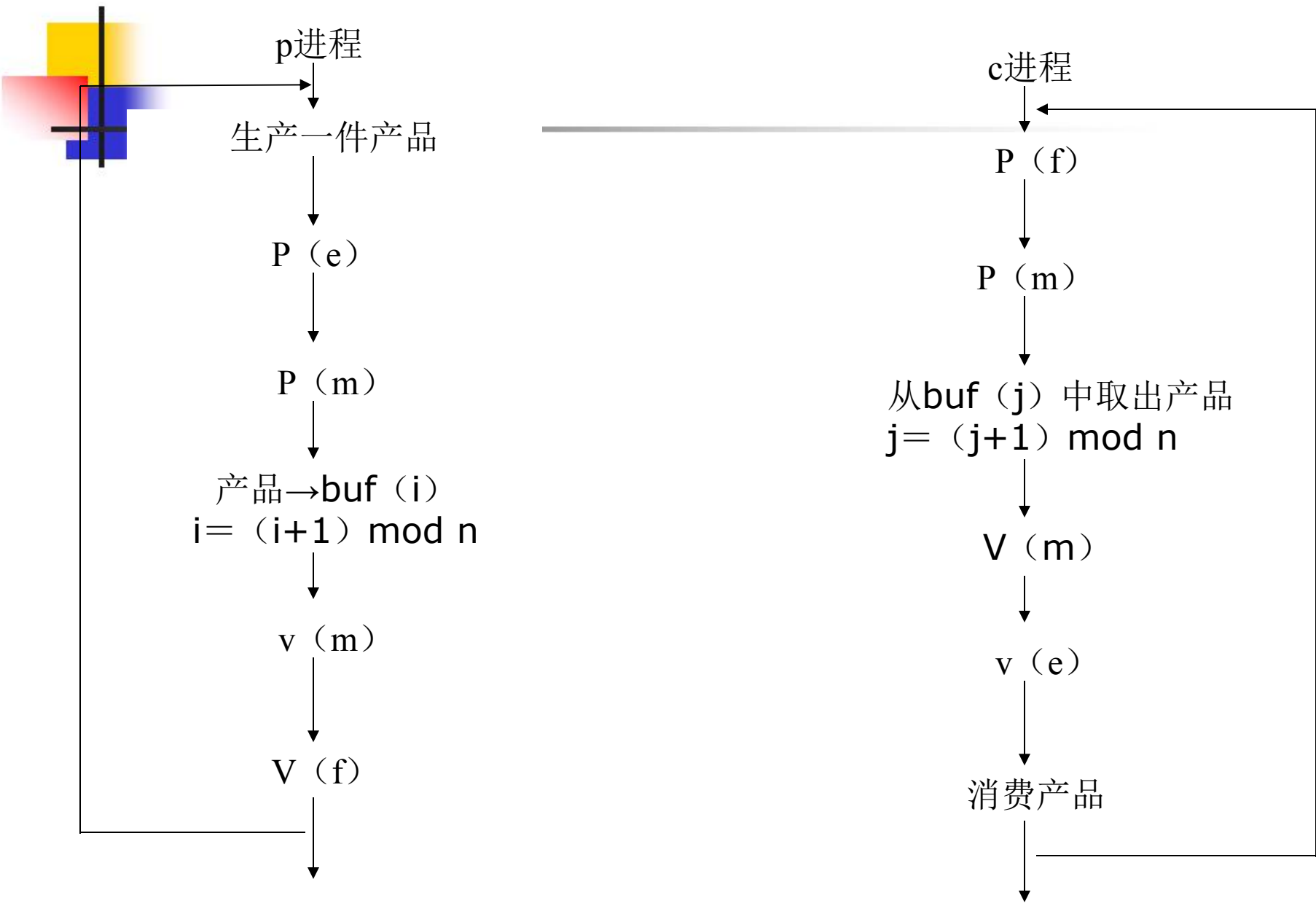
pc公用信号量 m : 互斥访问BUF

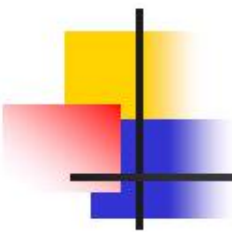
设: 指针 i 指向首空buf j 指针指向首产品

初值 $e=n$ $f=0$ $m=1$ $i=j=0$

$i=j$ BUF空

$(i+1) \bmod n = j$ BUF满





下述两段执行序列是否正确？请分析可能出现的问题，并说明理由。

① `wait (mutex);`
“临界段代码”；

网学天地 (www.e-studysky.com) 提供,
咨询QQ: 2696670126

`wait (mutex);`

② “临界段代码”；（没有对信号量的访问）

解答：

① 错误（1'）

分析：将 `signal (mutex)` 误写成 `wait (mutex)`. （2'）

后果：进程使用临界资源完毕后将无法释放资源，若该资源紧张则可能导致死锁，违背了空闲让进的原则. （2'）

② 错误（1'）

分析：`wait` 操作和 `signal` 操作缺失. （2'）

后果：进程将自由进入临界区使用临界资源，将导致资源被破坏的严重后果 （2'）

1.假定系统中有五个进程 {P0,P1,P2,P3,P4} 和三类资源 {A,B,C}，各种资源的数量分别为10，5，7，在T0时刻的资源分配情况：

资源 情况 进程	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	1	0	7	4	3	3	3	2
P1	3	2	2	2	0	0	1	2	2			
P2	9	0	2	3	0	2	6	0	0			
P3	2	2	2	2	1	1	0	1	1			
P4	4	3	3	0	0	2	4	3	1			

2. P1请求资源Request₁ (1, 0, 2)，系统按银行家算法进行检查：

① Request₁ (1, 0, 2) ≤ Need₁(1,2,2)

② Request₁ (1, 0, 2) ≤ Available₁(3,3,2)

资源 情况 进程	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	1	0	7	4	3	2	3	0
P1	3	2	2	3	0	2	0	2	0			
P2	9	0	2	3	0	2	6	0	0			
P3	2	2	2	2	1	1	0	1	1			
P4	4	3	3	0	0	2	4	3	1			

- 存在安全序列 {P1,P3,P4,P2,P0} ,因此，系统是安全的，可用立即将P1所申请的资源分配给他。

资源 情况 进程	Work			Need			Allocation			Work+Allocation			Finish
	A	B	C	A	B	C	A	B	C	A	B	C	
P													
P													
P													
P													
P													

(3) P4请求资源, Request₄ (3, 3, 0), 系统按银行家算法进行检查:

① Request₄ (3, 3, 0) ≤ Need₄(4,3,1)

② Request₄ (3, 3, 0) ≥ Available₄(2,3,0), 让P4等待。

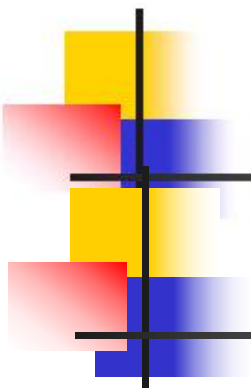
(4) P0请求资源, Request₀ (0, 2, 0), 系统按银行家算法进行检查:

① Request₀ (0, 2, 0) ≤ Need₀(7,4,3)

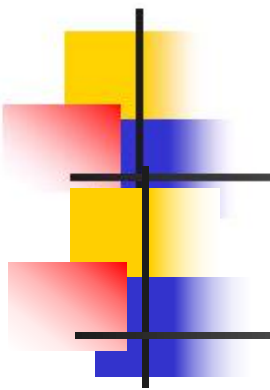
② Request₀ (0, 2, 0) ≤ Available₀(2,3,0)

③系统暂时先假定可为P0分配资源, 并修改数据。

	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	3	0	7	2	3	2	1	0
P1	3	2	2	3	0	2	0	2	0			
P2	9	0	2	3	0	2	6	0	0			
P3	2	2	2	2	1	1	0	1	1			
P4	4	3	3	0	0	2	4	3	1			

- 
- 2.某系统中四个进程的到达时间和要求服务时间如下表，请采用**SPF**（不抢占）调度算法进行分析，求进程执行序列和平均周转时间。要求有分析过程。

进程↵	到达时间↵	请求服务时间↵	↵
P1↵	0↵	3↵	↵
P2↵	1↵	6↵	↵
P3↵	3↵	3↵	↵
P4↵	4↵	1↵	↵

- 
- 3.考虑一个有**150**个存储器单元的系统，如下分配给三个进程：
 - 进程 最大需求 已分配
 -
 - | | | |
|---|----|----|
| 1 | 70 | 45 |
| 2 | 60 | 40 |
| 3 | 60 | 15 |
 - 使用银行家算法，以确定下面的任何一个请求是否安全：
 - a. 第4个进程到达，最多需要**60**个存储单元，最初需要**25**个单元；
 - b. 第4个进程到达，最多需要**60**个存储单元，最初需要**35**个单元；
 - 如果安全给出安全序列；若不安全给出结果分配简表。

a. (5 分)

进程	最大	占有	尚需	可用
1	70	45	25	25
2	60	40	20	
3	60	15	45	
4	60	25	35	

安全序列为：1、2、3、4

所以系统是安全的，可以进行分配。

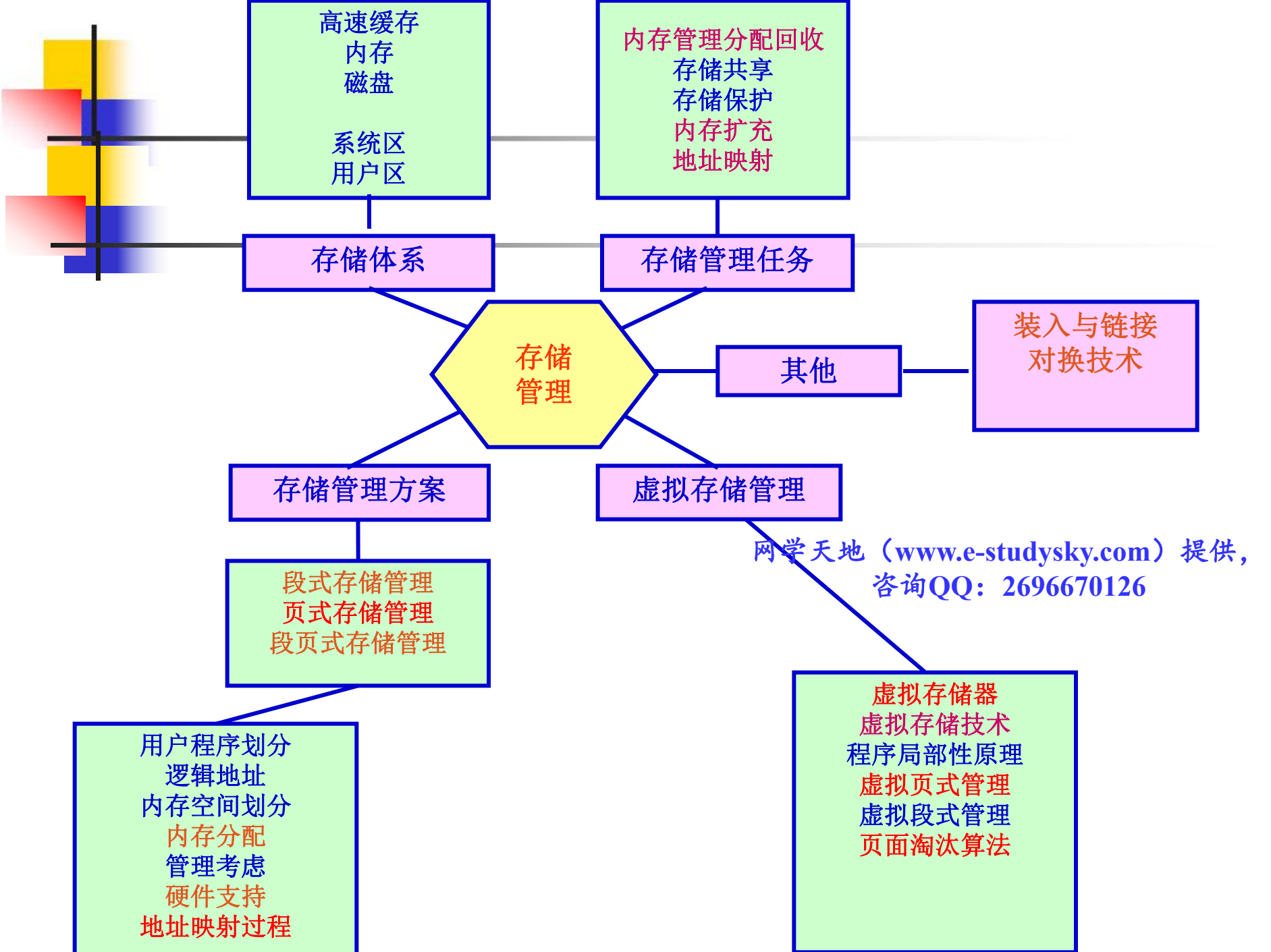
b. (5 分)

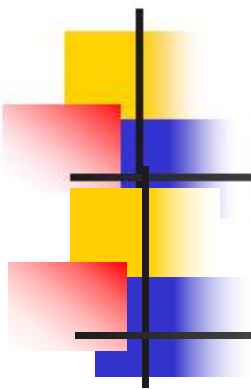
进程	最大	占有	尚需	可用
1	70	45	25	15
2	60	40	20	
3	60	15	45	
4	60	35	25	

当前可用的资源不够任何一个进程运行完毕，所以不安全。



•第四章 存储管理



- 
- 1、在一个请求分页系统中，假如系统分配给一个作业的物理块数为 4，且此作业的页面走向为：2，3，2，1，5，2，4，5，3，2，5，2。试用FIFO（先进先出）和OPT（最佳）两种算法分别计算出程序访问过程中所发生的缺页次数（假设开始执行时主存中没有页面，凡第一次用到的页面都产生一次缺页中断。要求列表计算）。

LRU?

3 ?

页面置换次数

1、解：（1）FIFO 方法，×表示缺页


FIFO	2	3	2	1	5	2	4	5	3	2	5	2
	2	3	3	1	5	5	4	4	4	2	2	2
		2	2	3	1	1	5	5	5	4	4	4
				2	3	3	1	1	1	5	5	5
					2	2	3	3	3	1	1	1
缺页	×	×	√	×	×	√	×	√	√	×	√	√

答：采用 FIFO 方法缺页次数为 6 次。（5 分）

（2）OPT 方法，×表示缺页

LRU	2	3	2	1	5	2	4	5	3	2	5	2
	2	3	3	1	5	5	4	4	4	4	4	4
		2	2	3	1	1	5	5	5	5	5	5
				2	3	3	3	3	3	3	3	3
					2	2	2	2	2	2	2	2
缺页	×	×	√	×	×	√	×	√	√	√	√	√

答：采用 OPT 方法缺页次数为 5 次。（5 分）



例2：已知某分页系统，主存容量为64K，页面大小为1K，对一个4页大的作业，其0、1、2、3页分别被分配到主存的2、4、6、7块中。

(1) 将十进制的逻辑地址1023、2500、3500、4500转换成物理地址？

(2) 以十进制的逻辑地址1023为例画出**地址变换过程图**？

答:

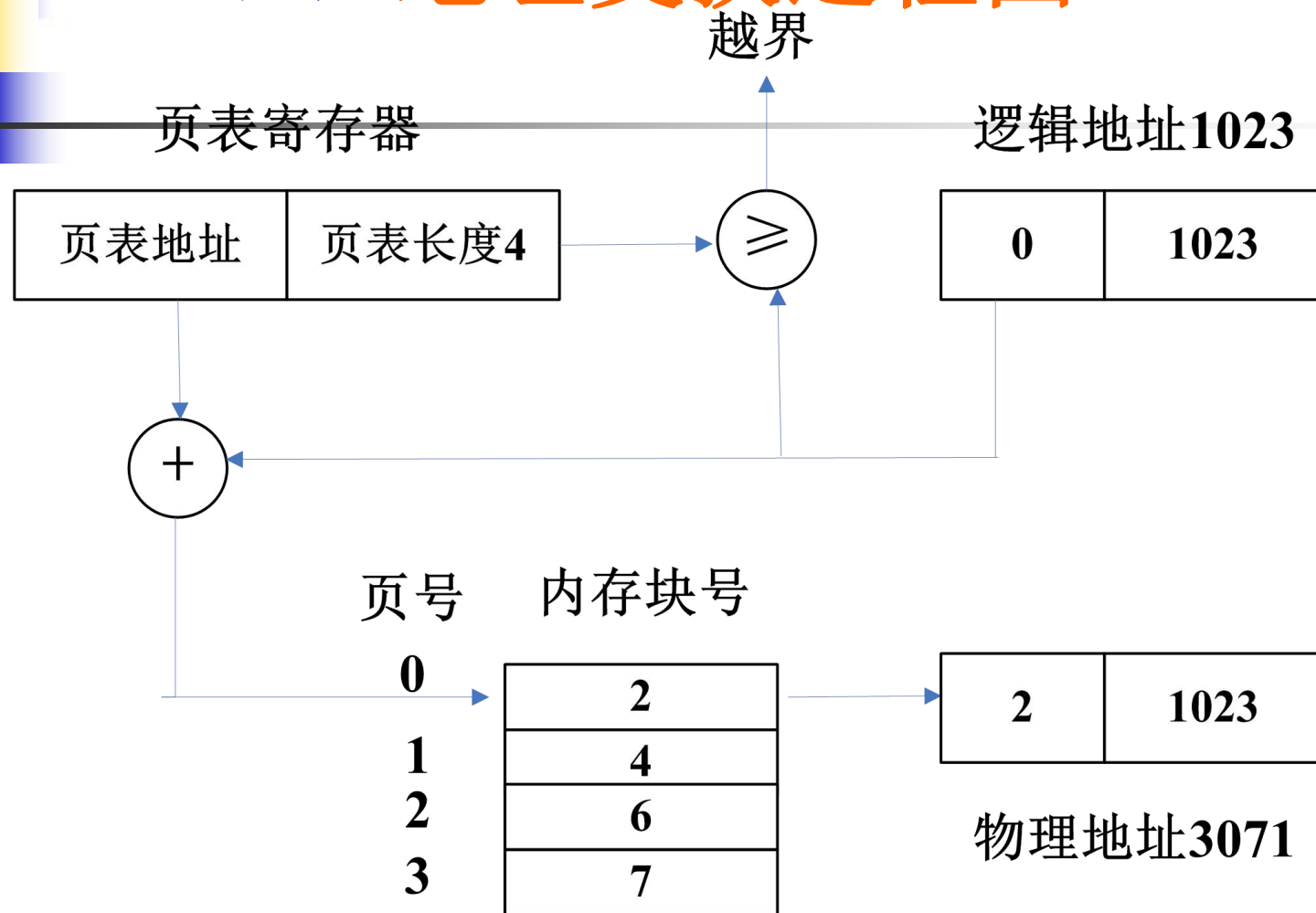
①逻辑地址1023: $1023/1K$, 得页号为0, 页内地址为1023, 查页表找到对应的物理块号为2, 故物理地址为 $2 \times 1K + 1023 = 3071$

②逻辑地址2500: $2500/1K$, 得页号为2, 页内地址为452, 查页表找到对应的物理块号为6, 故物理地址为 $6 \times 1K + 452 = 6596$

③逻辑地址3500: $3500/1K$, 得页号为3, 页内地址为428, 查页表找到对应的物理块号为7, 故物理地址为 $7 \times 1K + 428 = 7596$

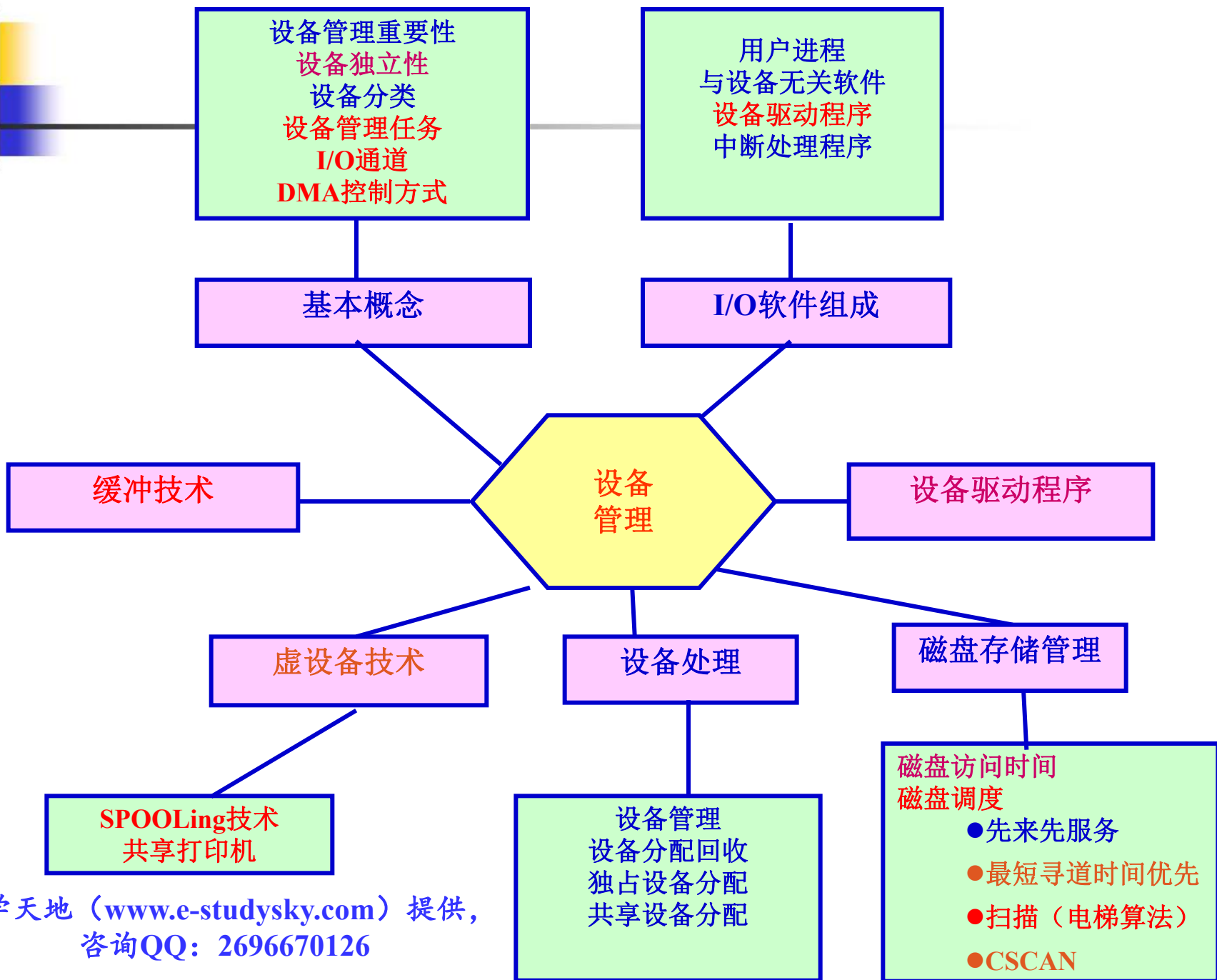
④逻辑地址4500: $4500/1K$, 得页号为4, 页内地址为404, 因页号不小于页表长度, 故产生越界中断。

(2) 地址变换过程图



方法 功能	单一 连续区	分区式		页式		段式	段页式
		固定	可变	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
重定位方式	静态	静态	动态	动态		动态	动态
分配方式	静态连续区	静态 动态连续区		静态或动态页为单位非连续		动态段为单位非连续	动态分配页为单位非连续
释放	执行完后全部释放	执行完后全部释放	分区释放	执行完后释放	淘汰与执行完后释放	淘汰与执行完后释放	淘汰与执行完后释放
保护	越界保护	越界保护与保护键		越界保护与控制权保护		越界保护与控制权保护	越界保护与控制权保护
内存扩充	覆盖与交换	覆盖与交换		覆盖交换	虚拟存储	虚拟存储	虚拟存储
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，重定位机构		地址变换机构，中断机构，保护机构		段式地址变换机构，保护与中断机构，动态连接机构	段式地址变换机构，保护与中断机构，动态连接机构

小结





• 第五章设备管理的重点、难点

- 设备管理的主要任务
- 什么叫通道技术
- 如何解决因通道不足而产生的瓶颈问题
- I/O 控制方式：四种I/O 方式的基本原理；I/O 方式由低到高效的演变的推动因素是什么？
- 缓冲的概念，为什么引入缓冲
- 中断处理程序的处理过程
- 设备分配方式



第五章设备管理的重点、难点

虚拟设备和SP00Ling 技术

- 什么是虚拟设备
- 什么是SP00Ling技术，SP00Ling系统的组成
- 如何利用SP00Ling技术实现共享打印机

磁盘调度

- 磁盘调度的目标
- 磁盘访问时间的计算
- FCFS、SSTF、SCAN、CSCAN 等算法的应用及这些调度算法的演变过程，分别解决了哪些问题；各算法的性能比较

例

一个磁盘系统，平均寻道时间为12ms，转速为10000转/分，每个磁道有18个扇区，每个扇区512个字节。请问要读取一个扇区所花的时间是多少？

解：

网学天地 (www.e-studysky.com) 提供,
咨询QQ: 2696670126

$$T_S = 12\text{ms}$$

$$T_R = 1/2r = 60 \div 10000 \times 0.5 = 3\text{ms}$$

$$T_A = b/rN = (512 \times 60) \div (18 \times 512 \times 10000) = 0.33\text{ms}$$

$$T_T = T_S + T_R + T_A = 12 + 3 + 0.33 = 15.33\text{ms}$$

答：读取一个扇区所花的时间是15.33ms。

磁盘调度

目标：减少寻道时间

1、FCFS (First Come First Served) 先来先服务

- 特点：公平、简单，寻道时间长，相当于随机访问模式。
- 仅适用于请求磁盘I/O的进程数目较少的场合。

2、SSTF (最短寻道优先) 最短寻道时间优先

- SSTF比FCFS有更好的寻道性能
- 贪心的算法
- 饥饿现象
- 不能保证平均寻道时间最短 ?



3、SCAN 扫描算法（也称为电梯算法）。

- 进程“饥饿现象”

SSTF存在。

- SCAN算法：

- 在移动方向固定的情况下采用了SSTF，以避免饥饿现象

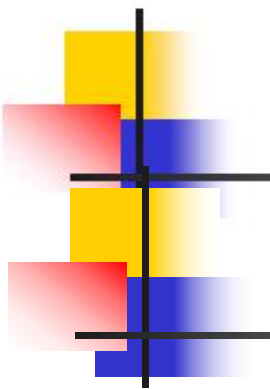
4、循环扫描CSCAN

- 磁头单向移动
- 一个方向读完，不是象SCAN那样回头，而是循环扫描。



例

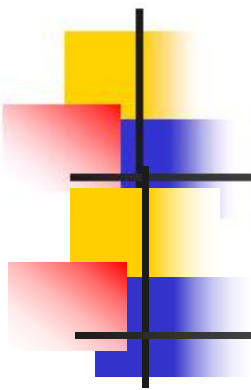
- 假设一个活动头磁盘有200道, 编号从0-199. 当前磁头正在143道上服务, 并且刚刚完成了125道的请求。现有如下访盘请求序列(磁道号): 86, 147, 91, 177, 94, 150, 102, 175, 130
- 试给出采用下列算法后磁头移动的顺序和移动总量(总磁道数).
- (1) 先来先服务(FCFS)磁盘调度算法. (2) 最短寻道时间优先(SSTF)磁盘调度算法.
- (3) 扫描法(SCAN)磁盘调度算法.(假设沿磁头移动方向不再有访问请求时, 磁头沿相反方向移动)

- 
- 答、
 - (1) 磁道访问顺序为: 86, 147, 91, 177, 94, 150, 102, 175, 130
 - 磁头移动的总磁道数 = $57+61+56+86+83+56+48+73+45 = 565$ (3分)
 - (2) 当前磁头在143道上:
 - 磁道访问顺序为: 147, 150, 130, 102, 94, 91, 86, 175, 177
 - 磁头移动的总磁道数 = $4+3+20+28+8+3+5+89+2 = 162$ (3分)
 - (3) 当前磁头在143道上, 并且刚刚完成125道的请求
 - 磁道访问顺序为: 147, 150, 175, 177, 130, 102, 94, 91, 86
 - 磁头移动的总磁道数 = $4+3+25+2+47+28+8+3+5 = 125$ (4分)



例

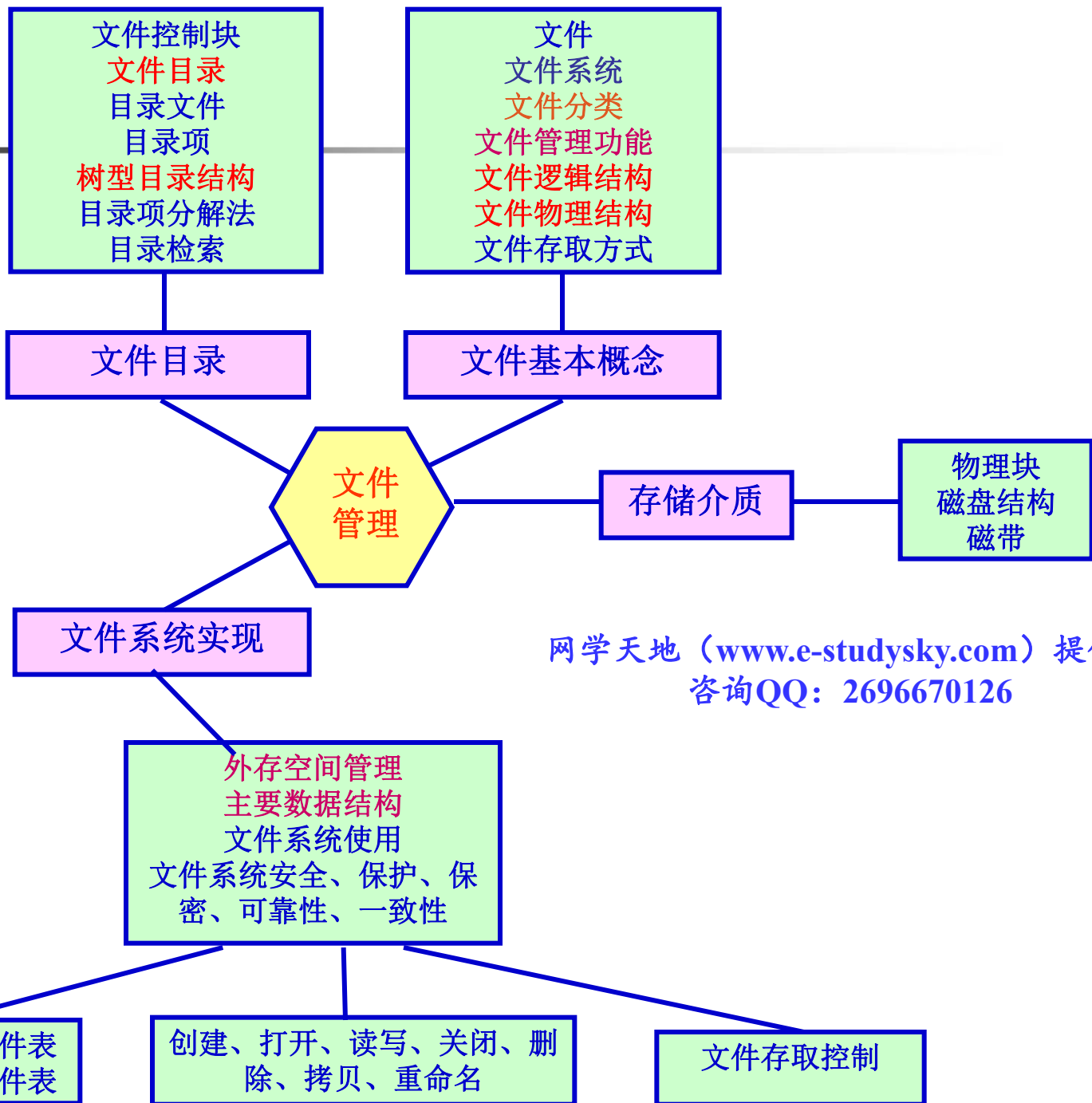
- 若干个等待访问磁盘者依次要访问的磁道为20, 44, 40, 4, 80, 12, 76, 假设每移动一个磁道需要3毫秒时间, 移动臂当前位于40号柱面, 请按下列算法分别写出访问序列并计算为完成上述各次访问总共花费的寻道时间。
- (1) 先来先服务算法; (2) 最短寻道时间优先算法。
- (3) 扫描算法 (当前磁头移动的方向为磁道递增)

- 
- 解：（1）磁道访问顺序为：20，44，40，4，80，12，76
 - 寻道时间 = $(20+24+4+36+76+68+64) * 3 = 292 * 3 = 876$ （3分）
 - （2）磁道访问顺序为：40，44，20，12，4，76，80
 - 寻道时间 = $(0+4+24+8+8+72+4) * 3 = 120 * 3 = 360$ （3分）
 - （3）磁道访问顺序为：40，44，76，80，20，12，4
 - 寻道时间 = $(0+4+32+4+60+8+8) * 3 = 116 * 3 = 348$ （4分）

磁盘访问时间由哪几部分组成？每部分时间应如何计算？

解：磁盘访问时间由下述部分组成：

- (1) 寻道时间 T_s ：指把磁臂(磁头)移动到指定磁道上所经历的时间。该时间是启动磁臂的时间 s 与磁头移动 n 条磁道所花费的时间之和，即 $T_s = m \times n + s$ 。(2分)
- (2) 旋转延迟时间 T_r ：这是指定扇区移动到磁头下面所经历的时间。(2分)
- (3) 传输时间 T_t ：从磁盘读出或写入 b 个字节数据所经历的时间。
- 其中， r 为磁盘每秒钟的转数； N 为一条磁道上的字节数。(3分)
- 因此，访问时间 T_a 表示为：(3分)



网学天地 (www.e-studysky.com) 提供,
咨询QQ: 2696670126

例

有一个计算机系统利用下图所示的位示图（行号、列号都从0开始编号）来管理空闲盘块。如果盘块从1开始编号，每个盘块的大小为1KB。

- (1) 现要从文件分配两盘块，试具体说明分配过程。
- (2) 若要释放磁盘的第300块，应如何处理？

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



解

(1) 分配过程

- 线形检索得: $i1=2$, $j1=2$; $i2=3$, $j2=6$ 。
- 计算空闲盘块号:
 - $b1=i1 \times 16+j1+1=2 \times 16+2+1=35$
 - $b2=i2 \times 16+j2+1=3 \times 16+6+1=55$
- 修改位示图:
 - 令 $map[2, 2]=map[3, 6]=1$, 并将对应块35, 55分配出去。



解

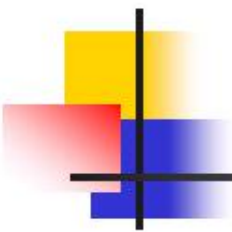
(2) 释放过程

- 计算出第300块所对应的二进制行号i和j
 - $i = (300 - 1) / 16 = 18$
 - $j = (300 - 1) \% 16 = 11$
- 修改位示图：
 - 令 $\text{map}[18, 11] = 0$ 。



例

- 某操作系统的磁盘文件空间共有**500**块，若用字长为**32**位的位示图管理磁盘空间，试问：
 - (1) 位示图需要多少字？
 - (2) 第*i*字第*j*位对应的块号是多少？

- 
-
- (1) 16个字。(5分)
 - (2) $b=(i-1)*32+j=32(i-1)+j$ (b从1开始计数, i, j也从1开始计数) (5分)