

## 第二章

1. 设有无穷多个缓冲区和无穷多个信息，A 进程把信息逐个的写入每个缓冲区，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)指出信号量的值的变化范围和其值的含义。

解：(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 := 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
    process B:
      begin
        j:=0;
        M: j:=j+1;
        P(S);
          读出第 j 个缓冲区信息;
        goto M;
      end
  parend;
end
```

(3) 信号量 S 的值域为 $[-1, \infty]$ 中的整数，当  $S=-1$  时，表示缓冲区从未写入或 B 读空，且 B 要求进一步读出，也即 B 超前 A 欲读取信息。

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

```
Var S, mutex1, mutex2: semaphore;
  S:=100; mutex1:=1; mutex2:=1
process Pi:
  begin
    P(S);
    P(mutex1);
    进入入口处，取一只篮子;
    V(mutex1);
    选购商品;
    P(mutex2);
```

结账，并归还篮子；

V(mutex2);

V(S);

End

3. 独木桥问题。某条河上只有一座独木桥，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。

Var S, S1, S2: semaphore;

rc1, rc2: integer;

S, S1, S2:=1;

rc1, rc2:=0;

process (E-W) i:

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:

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

4. 爸爸擀饼，妈妈烙饼，面板上只能容纳两张擀好的饼，只有当面板上有空时，爸爸才能把擀好的饼放在面板上，只有当面板上有饼时，妈妈才能从面板上取饼，试采用信号量机制实现爸爸与妈妈的同步。

解：设置两个信号量 var empty、full: semaphore。

初始：可以放饼的空位置数 empty=2，已经放饼的位置数 full=0

爸爸：

擀饼

wait(empty).

放饼

signal(full)

妈妈:

wait(full)

取饼

signal(empty)

烙饼

### 第三章

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(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、平均周转时间和平均带权周转时间。

(1) 先来先服务 FCFS 调度算法

周转时间=完成时间-提交时间

带权周转时间=周转时间/运行时间

作业号	提交时间	运行时间	开始时刻	完成时刻	周转时间	带权周转时间
1	10.00	2	10.00	12.00	2	1
2	10.10	1	12.00	13.00	2.9	2.9
3	10.25	0.25	13.00	13.25	3	12
平均					2.63	5.3

(2) 短作业优先 SJF 调度算法

作业号	提交时间	运行时间	开始时刻	完成时刻	周转时间	带权周转时间
1	10.00	2	10.00	12.00	2	1
2	10.10	1	12.25	13.25	3.15	3.15
3	10.25	0.25	12.00	12.25	2	8
平均					2.38	4.05

2. 设有四道作业，它们的提交时间和运行时间如下表：

作业号	提交时刻(时)	运行时间(小时)
1	8:00	2.0
2	8:50	0.5
3	9:00	0.1
4	9:50	0.2

求：试给出下面三种调度算法下，作业的执行顺序、平均周转时间和平均带权周转时间。

(1) 先来先服务 FCFS 调度算法

作业号	提交时间	运行时间	开始时刻	完成时刻	周转时间 (分钟)	带权周转时间
1	8:00	2.0	8:00	10:00	120	1

2	8:50	0.5	10:00	10:30	100	3.33
3	9:00	0.1	10:30	10:36	96	16
4	9:50	0.2	10:36	10:48	58	4.83
平均					1.647	6.29

(2) 短作业优先 SJF 调度算法

作业号	提交时间	运行时间	开始时刻	完成时刻	周转时间 (分钟)	带权周转 时间
1	8:00	2.0	8:00	10:00	120	1
2	8:50	0.5	10:18	10:48	118	3.93
3	9:00	0.1	10:00	10:06	66	11
4	9:50	0.2	10:06	10:18	28	2.33
平均					83	4.57

(3) 高响应比优先调度算法

优先权 = (等待时间 + 服务时间) / 服务时间

作业号	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	8:00	2.0	8:00	10:00	120	1.00
2	8:50	0.5	10:06	10:36	106	3.53
3	9:00	0.1	10:00	10:06	66	11.00
4	9:50	0.2	10:36	10:48	58	4.83
					1.375	5.09

2: (76+30)/0.5=212  
4: (16+12)/0.2=140

2: (70+30)/0.5=200  
3: (60+6)/0.1=660  
4: (10+12)/0.2=110

3. 银行家算法。

在银行家算法中，若出现下述资源分配情况，试问：

Process	Allocation	Need	Available
P0	0032	0012	1622
P1	1000	1750	
P2	1354	2356	
P3	0332	0652	
P4	0014	0656	

(1) 该状态是否安全？

解：该状态是安全的，因为存在一个安全序列<P0, P3, P4, P1, P2>。

下表为该时刻的安全序列：

进程	Work	Need	Allocation	Work+Allocation	Finish
P0	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	true
P3	1 6 5 4	0 6 5 2	0 3 3 2	1 9 8 6	true

P4	1 9 8 6	0 6 5 6	0 0 1 4	1 9 9 10	true
P1	1 9 9 10	1 7 5 0	1 0 0 0	2 9 9 10	true
P2	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, 系统能否将资源分配给它?

解: 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, 系统还剩的资源情况为 (0, 4, 0, 0), 此时系统中的资源将无法满足不同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。

4. 假设某系统中有 3 种资源 (R1, R2, R3), 在某时刻系统中共有 4 个进程, 进程 (P1, P2, P3, P4) 的最大资源需求数向量和此时已分配的资源数向量分别为:

进程	最大资源需求	当前已分配到资源
P1	3 2 2	1 0 0
P2	6 1 3	5 1 1
P3	3 1 4	2 1 1
P4	4 2 2	0 0 2

系统中当前可用资源向量为 (1, 1, 2), 问:

(1) 计算还需要资源数组;

	MAX	Allocation	Need	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) 系统此时是否安全?

	Work	Need	Allocation	Work+Allocation	Finish
P2	1 1 2	1 0 2	5 1 1	6 2 3	True
P3	6 2 3	1 0 3	2 1 1	8 3 4	True
P4	8 3 4	4 2 0	0 0 2	8 3 6	True
P1	8 3 6	2 2 2	1 0 0	9 3 6	True

存在安全序列 P2、P3、P4、P1, 即系统此时安全

(3) 如果进程 P2 发出资源请求向量 (1, 0, 1), 系统能否将资源分配给它?

①  $\text{Request}_2(1, 0, 1) \leq \text{Need}_2(1, 0, 2)$

②  $\text{Request}_2(1, 0, 1) \leq \text{Available}_2(1, 1, 2)$

	MAX	Allocation	Need	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	

利用安全性算法检测是否安全:

	Work	Need	Allocation	Work+Allocation	Finish
P2	0 1 1	0 0 1	6 1 2	6 2 3	True
P3	6 2 3	1 0 3	2 1 1	8 3 4	True
P4	8 3 4	4 2 0	0 0 2	8 3 6	True
P1	8 3 6	2 2 2	1 0 0	9 3 6	True

存在安全序列 P2、P3、P4、P1, 即系统此时安全

(4) 如果进程 P1 发出资源请求向量(1, 0, 1)，系统能否将资源分配给它？

①  $Request1(1, 0, 1) \leq Need1(2, 2, 2)$

②  $Request1(1, 0, 1) \leq Available1(1, 1, 2)$

	MAX	Allocation	Need	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	

进行安全性检查，可用资源 Available(0 1 1)已不能满足任何进程需要，故系统进入不安全状态。

5. 假设某系统中有 4 种资源，在某时刻系统中共有 5 个进程，进程(P0, P1, P2, P3, P4)的最大资源需求数向量和此时已分配的资源数向量分别为：

进程	最大资源需求	当前已分配到资源
P0	0 0 1 2	0 0 1 2
P1	2 7 5 0	2 0 0 0
P2	6 6 5 6	0 0 3 4
P3	4 3 5 6	2 3 5 4
P4	0 6 5 2	0 3 3 2

系统中当前可用资源向量为(2, 1, 0, 0)，问：

(1) 计算进程还需要请求的资源向量：

	MAX	Allocation	Need	Available
P0	0 0 1 2	0 0 1 2	0 0 0 0	2 1 0 0
P1	2 7 5 0	2 0 0 0	0 7 5 0	
P2	6 6 5 6	0 0 3 4	6 6 2 2	
P3	4 3 5 6	2 3 5 4	2 0 0 2	
P4	0 6 5 2	0 3 3 2	0 3 2 0	

(2) 系统当前是处于安全状态么？

	Work	Need	Allocation	Work+Allocation	Finish
P0	2 1 0 0	0 0 0 0	0 0 1 2	2 1 1 2	True
P3	2 1 1 2	2 0 0 2	2 3 5 4	4 4 6 6	True
P4	4 4 6 6	0 3 2 0	0 3 3 2	4 7 9 8	True
P1	4 7 9 8	0 7 5 0	2 0 0 0	6 7 9 8	True
P2	6 7 10 8	6 6 2 2	0 0 3 4	6 7 12 12	True

存在安全序列 P0、P3、P4、P1、P2，即系统此时安全

(2) 当进程 P2 申请(0, 1, 0, 0)时，系统能立即满足么？

①  $Request2(0, 1, 0, 0) \leq Need2(6, 6, 2, 2)$

②  $Request2(0, 1, 0, 0) \leq Available1(2, 1, 0, 0)$

	MAX	Allocation	Need	Available
P0	0 0 1 2	0 0 1 2	0 0 0 0	2 0 0 0
P1	2 7 5 0	2 0 0 0	0 7 5 0	
P2	6 6 5 6	0 1 3 4	6 5 2 2	
P3	4 3 5 6	2 3 5 4	2 0 0 2	
P4	0 6 5 2	0 3 3 2	0 3 2 0	

利用安全性算法检测是否安全：

	Work	Need	Allocation	Work+Allocation	Finish
P0	2 0 0 0	0 0 0 0	0 0 1 2	2 0 1 2	True
P3	2 0 1 2	2 0 0 2	2 3 5 4	4 3 6 6	True
P4	4 3 6 6	0 3 2 0	0 3 3 2	4 6 9 8	True
P1	4 6 9 8	0 7 5 0	2 0 0 0		False
P2		6 5 2 2	0 1 3 4		

不存在安全序列

## 第四、五章

1. 一个由 4 个页面(页号 0~3)，每页由 1024 个字节组成的程序，把它装入一个由 8 个物理块(块号 0~7)组成的存储器中，装入情况如表所示：

页号	物理块号
0	3
1	5
2	6
3	2

已知下面的逻辑地址(其中方括号中的第一个元素为页号，第二个元素为页内地址)，请按页表求出对应的物理地址。

- (1) [0, 100]  $3 \times 1024 + 100 = 3172$   
 (2) [1, 179]  $5 \times 1024 + 179 = 5299$   
 (3) [2, 785]  $6 \times 1024 + 785 = 6929$   
 (4) [3, 1010]  $2 \times 1024 + 1010 = 3058$

2. 设一个逻辑地址空间有 8 个页面，每页大小为 1024B，现将它映像到 32 块物理块的内存上。

试问： (1) 逻辑地址要用多少位表示：  $2^3 \times 2^{10}$  共 13 位  
 (2) 物理地址要用多少位表示：  $2^5 \times 2^{10}$  共 15 位

3. 某虚拟存储器的用户空间共 32 个页面，每页 1KB，主存 16KB。

试问： (1) 逻辑地址的有效位是 15 位；  
 (2) 物理地址需要 14 位；  
 (3) 假定某时刻系统为用户的第 0, 1, 2, 3 页分别分配的物理块号为 5, 10, 4, 7，试将逻辑地址 0A5CH 和 093CH 变换为物理地址。

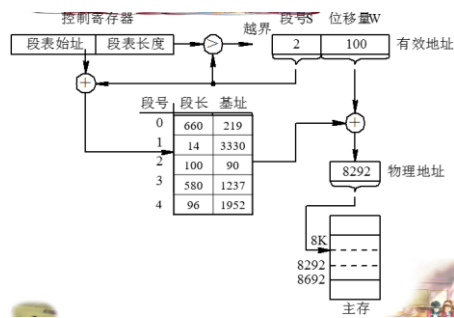
0000 1010 0101 1100                      0000 1001 0011 1100  
 0001 0010 0101 1100                      0001 0010 0101 1100  
 125CH    113CH

4. 某段式存储管理中采用如下表所示的段表。

段号	段的长度	内存起始地址
0	660	219
1	14	3330
2	100	90
3	580	1237
4	96	1952

试问：

- (1) 给定段号和段内地址，说明段式管理中的地址变换过程。



(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。

$[0, 430]$        $219+430=649$   
 $[1, 10]$        $3330+10=3340$   
 $[2, 500]$        $500>100$  段内地址越界  
 $[3, 400]$        $1237+400=1637$   
 $[4, 20]$        $1952+20=1972$   
 $[5, 100]$        $5>4$  段号越界

(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。

两次: 一次访问段表, 另一次是访问需要指令或数据。

5. 假定某请求页式虚拟系统中, 某进程的页面访问为:

0, 0, 3, 1, 1, 4, 0, 5, 6, 6, 2, 4, 6, 7, 7, 0, 0, 6, 7, 2, 进程实际页面数为 3, 则按先进先出 FIFO 置换算法和最近最久未使用 LRU 置换算法, 求缺页中断次数和缺页率。

FIFO:

	0	0	3	1	1	4	0	5	6	6	2	4	6	7	7	0	0	6	7	2
1	0	0	0	0	0	4	4	4	6	6	6	6	6	7	7	7	7	7	7	2
2			3	3	3	3	0	0	0	0	2	2	2	2	2	0	0	0	0	0
3				1	1	1	1	5	5	5	5	4	4	4	4	4	4	6	6	6
	调	命	调	调	命	替	替	替	替	命	替	替	命	替	命	替	命	调	命	替
	进	中	进	进	中	换	换	换	换	中	换	换	中	换	中	换	中	进	中	换

缺页中断次数: 13 次; 缺页率:  $13 \div 20 = 65\%$

LRU:

	0	0	3	1	1	4	0	5	6	6	2	4	6	7	7	0	0	6	7	2
1	0	0	0	0	0	4	4	4	6	6	6	6	6	6	6	6	6	6	6	6
2			3	3	3	3	0	0	0	0	2	2	2	7	7	7	7	7	7	7
3				1	1	1	1	5	5	5	5	4	4	4	4	0	0	0	0	2
	调	命	调	替	命	替	替	替	命	命	替	替	替	替	命	替	命	命	命	替
	进	中	进	换	中	换	换	换	中	中	换	换	换	换	中	换	中	中	中	换

缺页中断次数: 12 次; 缺页率:  $12 \div 20 = 60\%$

6. 在一个请求分页系统中, 假如一个进程的页面访问为: 4, 3, 2, 1, 4, 3, 5, 4, 3, 2, 1, 5, 当分配该进程的物理块数 M 分别为 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, 计算访问过程中所发生的缺页次数和缺页率。

FIFO M=3

	4	3	2	1	4	3	5	4	3	2	1	5
1	4	4	4	4	1	1	1	5	5	5	5	5
2		3	3	3	4	4	4	4	4	2	2	2



3			2	2	2	3	3	3	3	3	1	1
	调进	调进	调进	替换	替换	替换	替换	命中	命中	替换	替换	命中

缺页次数=9 次；缺页率=9/12=75%

LRU M=3

	4	3	2	1	4	3	5	4	3	2	1	5
1	4	4	4	1	1	1	5	5	5	2	2	2
2		3	3	3	4	4	4	4	4	4	1	1
3			2	2	2	3	3	3	3	3	3	5
	调进	调进	调进	替换	替换	替换	替换	命中	命中	替换	替换	替换

缺页次数=10 次；缺页率=10/12=83%

FIFO M=4

	4	3	2	1	4	3	5	4	3	2	1	5
1	4	4	4	4	4	4	5	5	5	5	1	5
2		3	3	3	3	3	3	4	4	4	4	4
3			2	2	2	2	2	2	3	3	3	3
4				1	1	1	1	1	1	2	2	2
	调进	调进	调进	调进	命中	命中	替换	替换	替换	替换	替换	替换

缺页次数=10 次；缺页率=10/12=83%

LRU M=4

	4	3	2	1	4	3	5	4	3	2	1	5
1	4	4	4	4	4	4	4	4	4	4	4	5
2		3	3	3	3	3	3	3	3	3	3	3
3			2	2	2	2	5	5	5	5	1	1
4				1	1	1	1	1	1	2	2	2
	调进	调进	调进	调进	命中	命中	替换	命中	命中	替换	替换	替换

缺页次数=8 次；缺页率=8/12=67%

## 第六章

1. 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)和 CSCAN(循环扫描)算法完成上述请求，写出磁头移动的顺序，并计算存取臂移动总量。

**FCFS(先来先服务)算法：**

143 道开始	
被访问的下一个磁道	移动距离
86	57
147	61
91	56
177	86
94	83

150	56
102	48
175	73
130	45
存取臂移动总量：565	

SSTF(最短寻道时间优先)算法：

143 道开始	
被访问的下一个磁道	移动距离
147	4
150	3
130	20
102	28
94	8
91	3
86	5
175	89
177	2
存取臂移动总量：162	

SCAN(扫描)算法：

143 道开始	
被访问的下一个磁道	移动距离
147	4
150	3
175	25
177	2
130	47
102	28
94	8
91	3
86	5
存取臂移动总量：125	

该算法不仅考虑到欲访问的磁道与当前磁道的距离，更优先考虑的是磁头的当前移动方向。这种算法中磁头移动的规律颇似电梯的运行，又称电梯调度算法。

CSCAN(循环扫描)算法：

143 道开始	
被访问的下一个磁道	移动距离
147	4
150	3
175	25
177	2

86	91
91	5
94	3
102	8
130	28
存取臂移动总量：169	

为了减少 SCAN 算法造成的某些进程的请求被严重推迟，CSCAN 算法规定磁头单向移动。