

## P1 进程的前驱后继问题

```
1 semaphore a,b,c,d,e,f,g,h;  
2 a=b=c=d=e=f=g=h=0;  
3 void S1(){S1 execute; signal(a); signal(b);}  
4 void S2(){wait(a); S2 execute; signal(c); signal(d);}  
5 void S3(){wait(b); S3 execute; signal(e);}  
6 void S4(){wait(c); S4 execute; signal(f);}  
7 void S5(){wait(d); S5 execute; signal(g);}  
8 void S6(){wait(e); S5 execute; signal(h);}  
9 void S7(){wait(f); wait(g); wait(h); S7 execute;}
```

## P2 银行家算法

(1) 系统处于安全状态，理由如下：

当前状态，进程的需求矩阵如下：

$$Need = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 7 & 5 & 0 \\ 2 & 3 & 5 & 6 \\ 0 & 6 & 5 & 2 \\ 0 & 6 & 5 & 6 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 3 & 5 & 4 \\ 0 & 6 & 3 & 2 \\ 0 & 0 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 6 & 4 & 2 \end{bmatrix} \quad (1)$$

银行家算法计算如下：

|      | A | B | C | D |            | A | B | C | D |  |  |  |  |
|------|---|---|---|---|------------|---|---|---|---|--|--|--|--|
| p1   | 0 | 0 | 0 | 0 | p1         | 0 | 0 | 1 | 2 |  |  |  |  |
| p2   | 0 | 7 | 5 | 0 | p2         | 1 | 0 | 0 | 0 |  |  |  |  |
| p3   | 1 | 0 | 0 | 2 | p3         | 1 | 3 | 5 | 4 |  |  |  |  |
| p4   | 0 | 0 | 2 | 0 | p4         | 0 | 6 | 3 | 2 |  |  |  |  |
| p5   | 0 | 6 | 4 | 2 | p5         | 0 | 0 | 1 | 4 |  |  |  |  |
| Need |   |   |   |   | Allocation |   |   |   |   |  |  |  |  |
|      |   |   |   |   |            |   |   |   |   |  |  |  |  |

| A         | B | C | D |
|-----------|---|---|---|
| 1         | 5 | 2 | 0 |
| Available |   |   |   |

get p1

|      | A | B | C | D |            | A | B | C | D |  |  |  |  |
|------|---|---|---|---|------------|---|---|---|---|--|--|--|--|
| p1   | 0 | 0 | 0 | 0 | p1         | 0 | 0 | 0 | 0 |  |  |  |  |
| p2   | 0 | 7 | 5 | 0 | p2         | 1 | 0 | 0 | 0 |  |  |  |  |
| p3   | 1 | 0 | 0 | 2 | p3         | 1 | 3 | 5 | 4 |  |  |  |  |
| p4   | 0 | 0 | 2 | 0 | p4         | 0 | 6 | 3 | 2 |  |  |  |  |
| p5   | 0 | 6 | 4 | 2 | p5         | 0 | 0 | 1 | 4 |  |  |  |  |
| Need |   |   |   |   | Allocation |   |   |   |   |  |  |  |  |
|      |   |   |   |   |            |   |   |   |   |  |  |  |  |

| A         | B | C | D |
|-----------|---|---|---|
| 1         | 5 | 3 | 2 |
| Available |   |   |   |

get p3

|      | A | B | C | D |            | A | B | C | D |  |  |  |  |
|------|---|---|---|---|------------|---|---|---|---|--|--|--|--|
| p1   | 0 | 0 | 0 | 0 | p1         | 0 | 0 | 0 | 0 |  |  |  |  |
| p2   | 0 | 7 | 5 | 0 | p2         | 1 | 0 | 0 | 0 |  |  |  |  |
| p3   | 0 | 0 | 0 | 0 | p3         | 0 | 0 | 0 | 0 |  |  |  |  |
| p4   | 0 | 0 | 2 | 0 | p4         | 0 | 6 | 3 | 2 |  |  |  |  |
| p5   | 0 | 6 | 4 | 2 | p5         | 0 | 0 | 1 | 4 |  |  |  |  |
| Need |   |   |   |   | Allocation |   |   |   |   |  |  |  |  |
|      |   |   |   |   |            |   |   |   |   |  |  |  |  |

| A         | B | C | D |
|-----------|---|---|---|
| 2         | 8 | 8 | 6 |
| Available |   |   |   |

get p4

|      | A | B | C | D |            | A | B | C | D |  |  |  |  |
|------|---|---|---|---|------------|---|---|---|---|--|--|--|--|
| p1   | 0 | 0 | 0 | 0 | p1         | 0 | 0 | 0 | 0 |  |  |  |  |
| p2   | 0 | 7 | 5 | 0 | p2         | 1 | 0 | 0 | 0 |  |  |  |  |
| p3   | 0 | 0 | 0 | 0 | p3         | 0 | 0 | 0 | 0 |  |  |  |  |
| p4   | 0 | 0 | 0 | 0 | p4         | 0 | 0 | 0 | 0 |  |  |  |  |
| p5   | 0 | 6 | 4 | 2 | p5         | 0 | 0 | 1 | 4 |  |  |  |  |
| Need |   |   |   |   | Allocation |   |   |   |   |  |  |  |  |
|      |   |   |   |   |            |   |   |   |   |  |  |  |  |

| A         | B  | C  | D |
|-----------|----|----|---|
| 2         | 14 | 11 | 8 |
| Available |    |    |   |

get p2

|      | A | B | C | D |            | A | B | C | D |  |  |  |  |
|------|---|---|---|---|------------|---|---|---|---|--|--|--|--|
| p1   | 0 | 0 | 0 | 0 | p1         | 0 | 0 | 0 | 0 |  |  |  |  |
| p2   | 0 | 0 | 0 | 0 | p2         | 0 | 0 | 0 | 0 |  |  |  |  |
| p3   | 0 | 0 | 0 | 0 | p3         | 0 | 0 | 0 | 0 |  |  |  |  |
| p4   | 0 | 0 | 0 | 0 | p4         | 0 | 0 | 0 | 0 |  |  |  |  |
| p5   | 0 | 6 | 4 | 2 | p5         | 0 | 0 | 1 | 4 |  |  |  |  |
| Need |   |   |   |   | Allocation |   |   |   |   |  |  |  |  |
|      |   |   |   |   |            |   |   |   |   |  |  |  |  |

| A         | B  | C  | D |
|-----------|----|----|---|
| 3         | 14 | 11 | 8 |
| Available |    |    |   |

get p5

|      | A | B | C | D |            | A | B | C | D |  |  |  |  |
|------|---|---|---|---|------------|---|---|---|---|--|--|--|--|
| p1   | 0 | 0 | 0 | 0 | p1         | 0 | 0 | 0 | 0 |  |  |  |  |
| p2   | 0 | 0 | 0 | 0 | p2         | 0 | 0 | 0 | 0 |  |  |  |  |
| p3   | 0 | 0 | 0 | 0 | p3         | 0 | 0 | 0 | 0 |  |  |  |  |
| p4   | 0 | 0 | 0 | 0 | p4         | 0 | 0 | 0 | 0 |  |  |  |  |
| p5   | 0 | 0 | 0 | 0 | p5         | 0 | 0 | 0 | 0 |  |  |  |  |
| Need |   |   |   |   | Allocation |   |   |   |   |  |  |  |  |
|      |   |   |   |   |            |   |   |   |   |  |  |  |  |

| A         | B  | C  | D  |
|-----------|----|----|----|
| 3         | 14 | 12 | 12 |
| Available |    |    |    |

[illegible]

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 0 | 3 | 3 | 0 |
| p3 | 1 | 0 | 0 | 2 |
| p4 | 0 | 0 | 2 | 0 |
| p5 | 0 | 6 | 4 | 2 |

Need

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 1 | 2 |
| p2 | 1 | 4 | 2 | 0 |
| p3 | 1 | 3 | 5 | 4 |
| p4 | 0 | 6 | 3 | 2 |
| p5 | 0 | 0 | 1 | 4 |

Allocation

| A | B | C | D |
|---|---|---|---|
| 1 | 1 | 0 | 0 |

Available

get p1

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 0 | 3 | 3 | 0 |
| p3 | 1 | 0 | 0 | 2 |
| p4 | 0 | 0 | 2 | 0 |
| p5 | 0 | 6 | 4 | 2 |

Need

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 1 | 4 | 2 | 0 |
| p3 | 1 | 3 | 5 | 4 |
| p4 | 0 | 6 | 3 | 2 |
| p5 | 0 | 0 | 1 | 4 |

Allocation

| A | B | C | D |
|---|---|---|---|
| 1 | 1 | 1 | 2 |

Available

get p3

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 0 | 3 | 3 | 0 |
| p3 | 0 | 0 | 0 | 0 |
| p4 | 0 | 0 | 2 | 0 |
| p5 | 0 | 6 | 4 | 2 |

Need

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 1 | 4 | 2 | 0 |
| p3 | 0 | 0 | 0 | 0 |
| p4 | 0 | 6 | 3 | 2 |
| p5 | 0 | 0 | 1 | 4 |

Allocation

| A | B | C | D |
|---|---|---|---|
| 2 | 4 | 6 | 6 |

Available

get p4

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 0 | 3 | 3 | 0 |
| p3 | 0 | 0 | 0 | 0 |
| p4 | 0 | 0 | 0 | 0 |
| p5 | 0 | 6 | 4 | 2 |

Need

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 1 | 4 | 2 | 0 |
| p3 | 0 | 0 | 0 | 0 |
| p4 | 0 | 0 | 0 | 0 |
| p5 | 0 | 0 | 1 | 4 |

Allocation

| A | B  | C | D |
|---|----|---|---|
| 2 | 10 | 9 | 8 |

Available

get p2

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 0 | 0 | 0 | 0 |
| p3 | 0 | 0 | 0 | 0 |
| p4 | 0 | 0 | 0 | 0 |
| p5 | 0 | 6 | 4 | 2 |

Need

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 0 | 0 | 0 | 0 |
| p3 | 0 | 0 | 0 | 0 |
| p4 | 0 | 0 | 0 | 0 |
| p5 | 0 | 0 | 1 | 4 |

Allocation

| A | B  | C  | D |
|---|----|----|---|
| 3 | 14 | 11 | 8 |

Available

get p5

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 0 | 0 | 0 | 0 |
| p3 | 0 | 0 | 0 | 0 |
| p4 | 0 | 0 | 0 | 0 |
| p5 | 0 | 0 | 0 | 0 |

Need

|    | A | B | C | D |
|----|---|---|---|---|
| p1 | 0 | 0 | 0 | 0 |
| p2 | 0 | 0 | 0 | 0 |
| p3 | 0 | 0 | 0 | 0 |
| p4 | 0 | 0 | 0 | 0 |
| p5 | 0 | 0 | 0 | 0 |

Allocation

| A | B  | C  | D  |
|---|----|----|----|
| 3 | 14 | 12 | 12 |

Available

## P3 处理器调度算法

(1)

FCFS调度算法（非抢占）的平均周转时间为：1.48 s，调度顺序按照提交时间顺序，为1→2→3→4，计算如下：

| FCFS调度算法（非抢占） |       |      |       |       |      |
|---------------|-------|------|-------|-------|------|
| 作业号           | 提交时间  | 运行时间 | 开始时间  | 结束时间  | 周转时间 |
| 1             | 10.00 | 1    | 10.00 | 11.00 | 1.00 |
| 2             | 10.10 | 0.6  | 11.00 | 11.60 | 1.50 |
| 3             | 10.20 | 0.3  | 11.60 | 11.90 | 1.70 |
| 4             | 10.30 | 0.1  | 11.90 | 12.00 | 1.70 |
| 平均周转时间        |       |      |       |       | 1.48 |

最短作业优先算法（非抢占）的平均周转时间为：1.30 s，调度顺序按照就绪队列中的运行时间顺序，为1→4→3→2，计算如下：

| 最短作业优先算法（非抢占） |       |      |       |       |      |
|---------------|-------|------|-------|-------|------|
| 作业号           | 提交时间  | 运行时间 | 开始时间  | 结束时间  | 周转时间 |
| 1             | 10.00 | 1    | 10.00 | 11.00 | 1.00 |
| 4             | 10.30 | 0.1  | 11.00 | 11.10 | 0.80 |
| 3             | 10.20 | 0.3  | 11.10 | 11.40 | 1.20 |
| 2             | 10.10 | 0.6  | 11.40 | 12.00 | 1.90 |
| 平均周转时间        |       |      |       |       | 1.30 |

(2) 当CPU使用响应比高者优先算法时，第一个作业完成后，t=11s，后续作业序列按照响应比顺序，为4→3→2。计算过程如下：

| 响应比高者优先算法 第一个作业完成后，t=11 |       |      |      |      |
|-------------------------|-------|------|------|------|
| 作业号                     | 提交时间  | 运行时间 | 等待时间 | 响应比  |
| 2                       | 10.10 | 0.6  | 0.90 | 2.50 |
| 3                       | 10.20 | 0.3  | 0.80 | 3.67 |
| 4                       | 10.30 | 0.1  | 0.70 | 8.00 |
| 平均周转时间                  |       |      |      |      |

## P4 分页存储

由题意得到，页表大小为1024B，分页存储的内存模型示意图如下：

|   |     |      |
|---|-----|------|
| 0 |     | 0    |
| 1 |     | 1024 |
| 2 | 页面1 | 2048 |
| 3 | 页面0 | 3072 |
| 4 | 页面3 | 4096 |

其中页大小为  $1024B = 2^{10}B$ ，故二进制虚地址中，0~9位为页内偏移量，10~31位为页面号。

虚地址 1044 ==> 二进制 1 0000010100，页面号为1，页内偏移量为 10100，即20。其中页面1对应2号存储块，起始地址为2048。

故实际物理地址为：2048+20=2068。

## P5 页面置换算法

(1) 最近最少使用 (LRU) 算法：

在2块主存空间中，缺页中断18次

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

在4块主存空间中，缺页中断10次

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

(3) 先进先出 (FIFO) 调度算法：

在2块主存空间中，缺页中断18次

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

在4块主存空间中，缺页中断14次

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