

# 第六章

## 6.5 银行家算法

a.

$A_r = 15 - (2 + 4 + 1 + 1 + 1) = 6$

$B_r = 6 - (1 + 1 + 1) = 3$

$C_r = 9 - (2 + 1 + 1) = 5$

$D_r = 10 - (1 + 1 + 2 + 1 + 1) = 4$

可验证正确性

b.

要求需求矩阵，使用最大需求减去当前已分配即可

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
$P_0$	7	5	3	4
$P_1$	2	1	2	2
$P_2$	3	4	4	2
$P_3$	2	3	3	1
$P_4$	4	1	2	1
$P_5$	3	4	3	3

c.

给出两种序列参考

序列1

进程	进程终止时可用资源
$P_1$	[6, 4, 6, 5]
$P_2$	[10, 5, 6, 7]
$P_0$	[12, 5, 8, 8]
$P_3$	[13, 5, 8, 9]
$P_4$	[14, 6, 8, 9]

$P_5$	[15, 6, 9, 10]
-------	----------------

序列2

进程	进程终止时可用资源
$P_1$	[6, 4, 6, 5]
$P_4$	[7, 5, 6, 5]
$P_5$	[8, 5, 7, 6]
$P_2$	[12, 6, 7, 8]
$P_3$	[13, 6, 7, 9]
$P_0$	[15, 6, 9, 10]

**d.**

不应

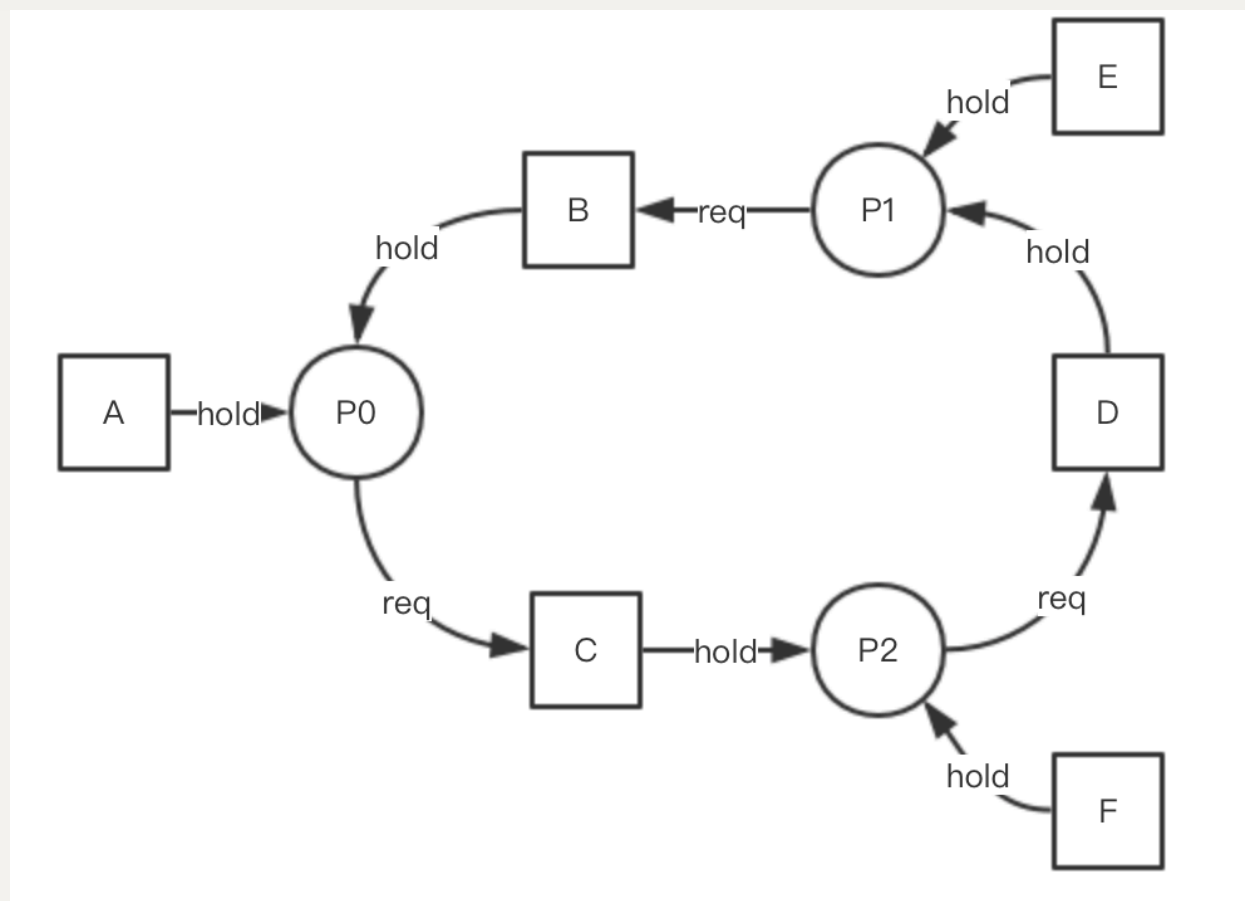
此时需求矩阵为

	$A$	$B$	$C$	$D$
$P_0$	7	5	3	4
$P_1$	2	1	2	2
$P_2$	3	4	4	2
$P_3$	2	3	3	1
$P_4$	4	1	2	1
$P_5$	0	2	0	0

假设允许 $P_5$ 的请求，此时可用资源向量为[3, 1, 2, 1]，若只有进程获取所有资源之后才会释放资源，此时可用资源向量不满足需求矩阵中任何一行，故发生死锁

## 6.6

**a.**



使用方格表示资源，圆圈表示进程，资源指向进程表示资源被某一个进程占有，进程指向资源表示进程请求某一个资源。资源A、B、C、D、E、F仅拥有一个单元的资源，此时资源图成环，会发生死锁。

**b.**

只需打破**a.**中资源图循环即可，例如：

P0	P1	P2
B	E	C
C	B	F
A	D	D

## 6.7

当 I 的速度远大于 O 的时候，例如当  $I=\max$ ，而  $O=0$  的时候，就会导致磁盘中都是输入的块，那么这个时候就会导致：

- I 等待空间来输入
- P 等待空间来输出

由于磁盘不能得到输出，所以 O 进程不会消耗它，导致死锁。

## 6.14

**a.**

会，foo运行semwait(S)，bar运行semwait(R)，两个进程分别在semwait(R)、semwait(S)阻塞。

**b.**

不会，两个进程同时运行，foo在semsignal(R)后，bar在semwait(R)的阻塞被解除，反之亦然，故不会出现有一个被无限延后。