

死锁

死锁

原因及必要条件

处理问题

预防死锁

破坏必要条件

避免死锁

银行家算法

检测与解除死锁

死锁定理+资源分配图

- 三胞胎
 - 死锁：资源被占有，双方互不相让，导致都不能运行。
 - 活锁：双方互相谦让，导致资源空闲。
 - 饥饿：有一方等待时间过长。
- 死锁避免
 - 银行家算法

安全序列（不唯一）

找到

无死锁

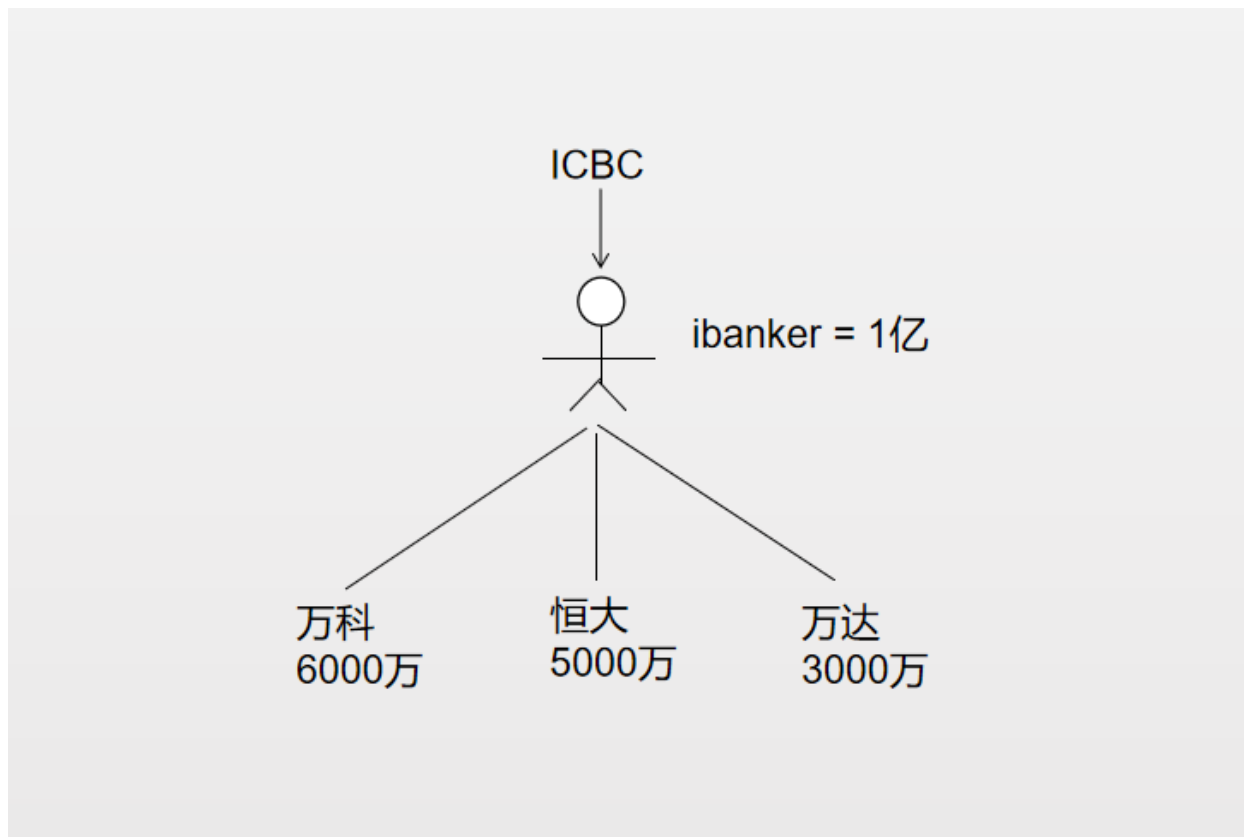
可避免

找不到

必死锁

无法避免

- 银行家算法案例：



	最大值	已分配	可需要
万科	6000	4000	2000
恒大	5000	3000	2000
万达	3000	2000	1000

一共有资产一亿，已经分配的有 $4000 + 3000 + 2000 = 9000$ 万，还剩1000万，可以满足的只有万达，所以将1000万分配给万达。

	最大值	已分配	可需要
万科	6000	4000	2000
恒大	5000	3000	2000

因为上面将剩余的已分配给万达，所以万达运行完，就会释放资源，所以目前剩下3000万的资产，可以将其分配给万科或者恒大，我就将其资产分配给万科。剩余 $3000 - 2000 = 1000$ 万。

	最大值	已分配	可需要
恒大	5000	3000	2000

将万达释放的资源分配给万科之后，万科运行完就会释放资源，目前资产就会剩余 $1000 + 6000 = 7000$ 万，最后将其分配给恒大。

所以安全序列为 万达-万科-恒大 或者 万达-恒大-万科。

- 案例：
 - available (拥有的)
 - Max (最大分配量)
 - allocation (已分配)
 - need (可需要)
 - residue (剩余量)

$$available = [3 \quad 3 \quad 2] \quad (1)$$

$$Max = \begin{bmatrix} 7 & 5 & 3 \\ 3 & 2 & 2 \\ 9 & 0 & 2 \\ 2 & 2 & 2 \\ 4 & 3 & 3 \end{bmatrix} \quad (2)$$

$$allocation = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 2 \\ 2 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \quad (3)$$

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

$$need = Max - allocation$$

上述对应的线程是P1, P2, P3, P4, P5, 拥有的资源量是A,B,C。

- 第一步：上述进程中，满足的有P2进程。释放P2进程资源，计算公式 $available - need + Max$ 。

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

$$residue = [5 \quad 3 \quad 2] \quad (6)$$

- 第二步，满足的有P4和P5进程，这里以P4进程为例。

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

$$residue = [7 \quad 4 \quad 3] \quad (8)$$

- 第三步，满足的有P1进程。

$$need = \begin{bmatrix} 6 & 0 & 0 \\ 4 & 3 & 1 \end{bmatrix} \quad (5)$$

$$residue = [7 \quad 5 \quad 3] \quad (8)$$

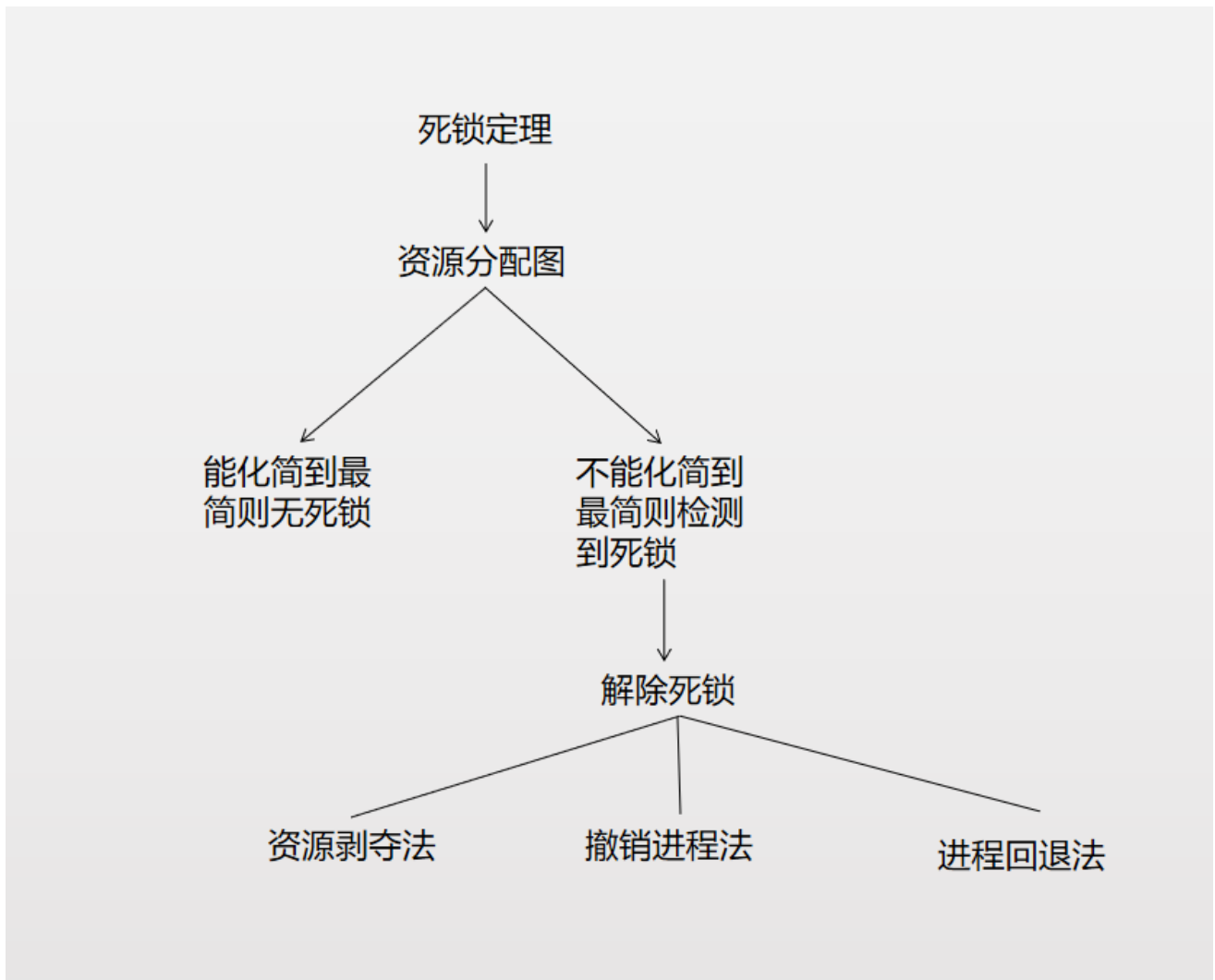
- 第四步，满足的有P3进程。

$$need = [4 \quad 3 \quad 1] \quad (5)$$

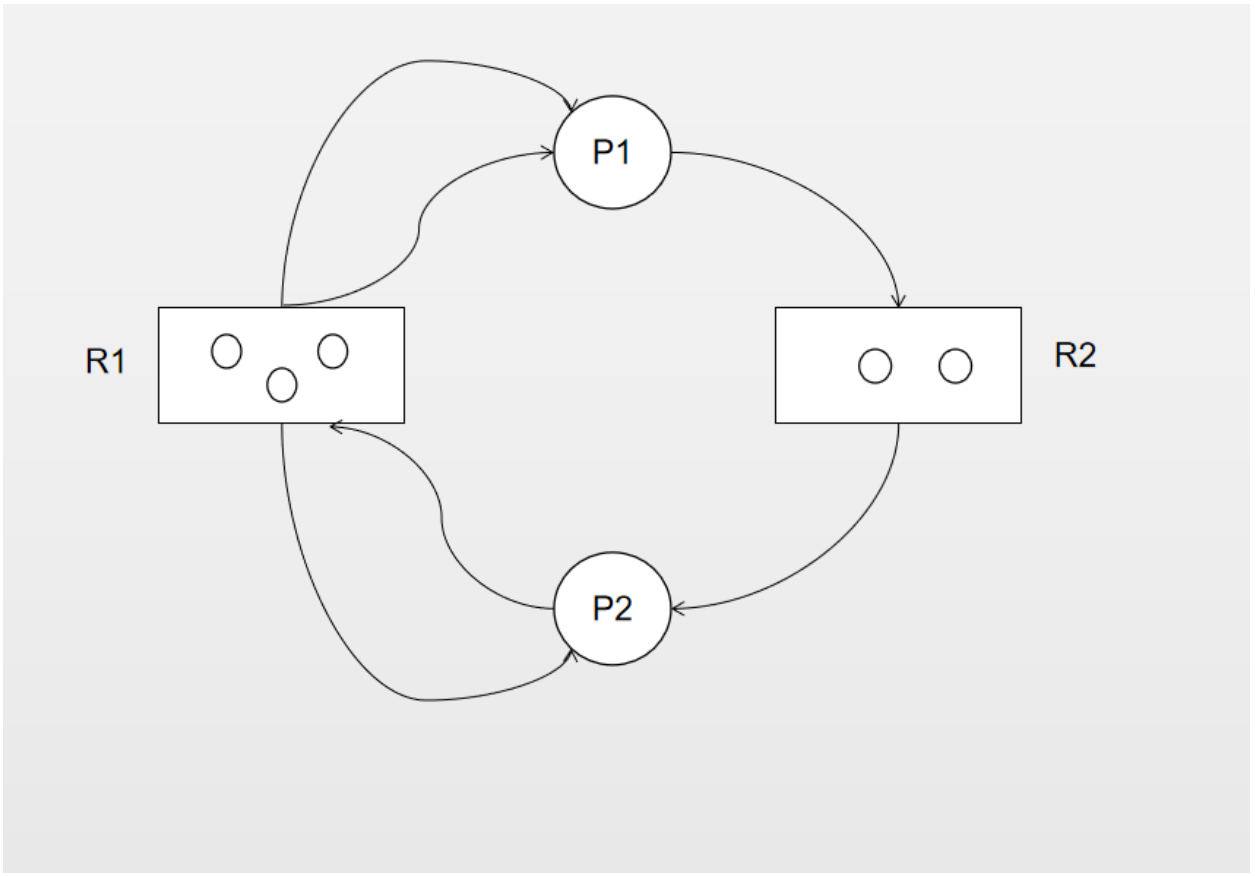
$$residue = [10 \quad 5 \quad 5] \quad (8)$$

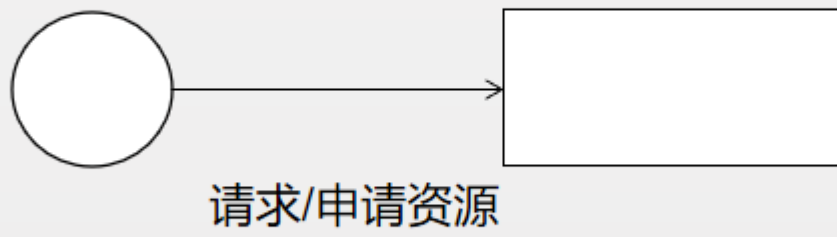
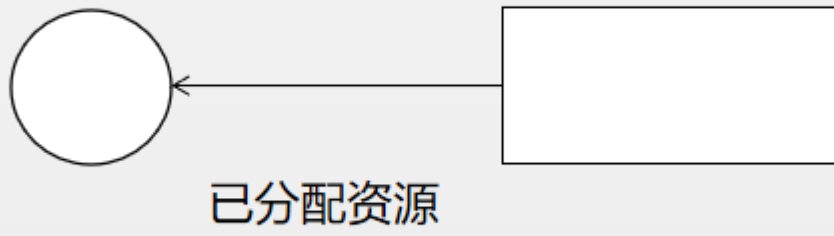
- 第五步，进程结束。

此程序的安全序列是P2, P4, P1, P3, P5。

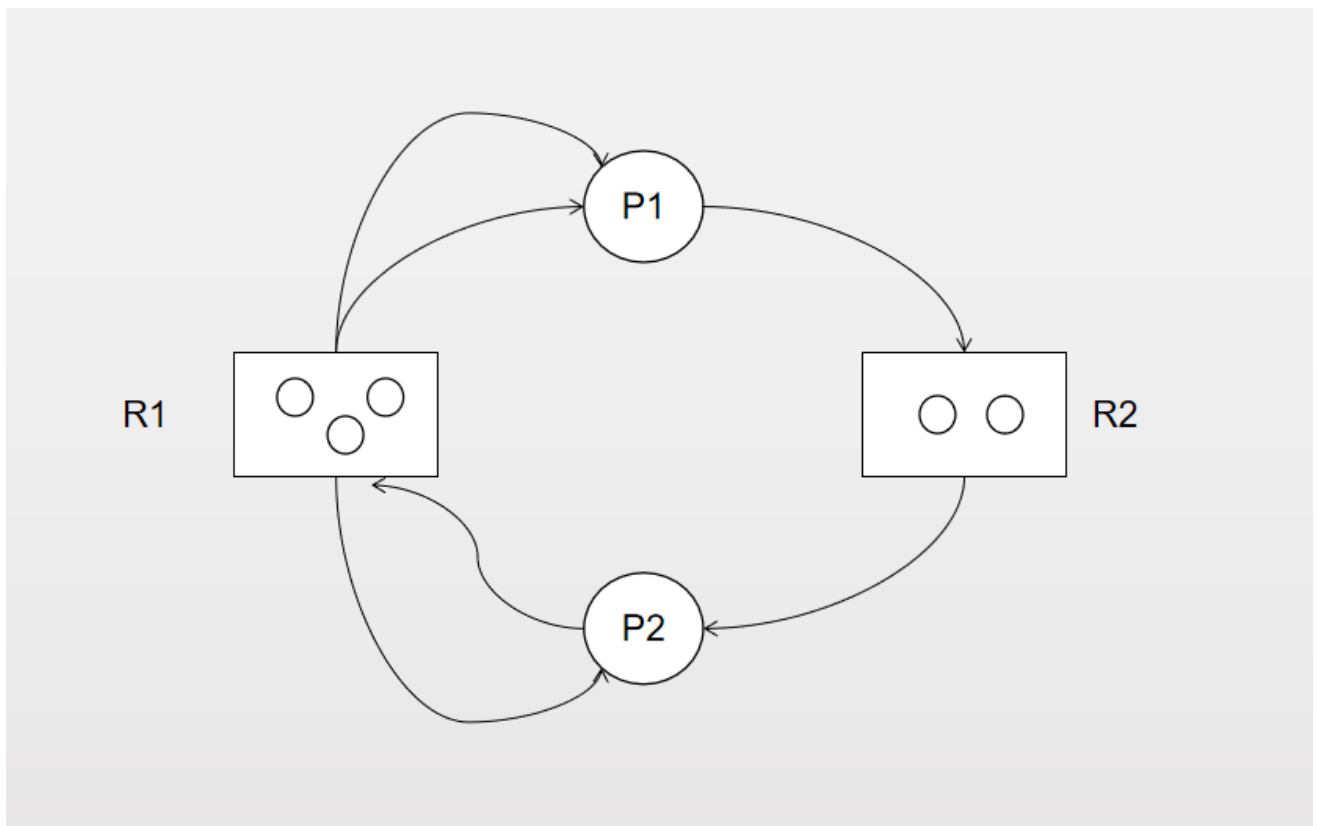


- 资源转换图

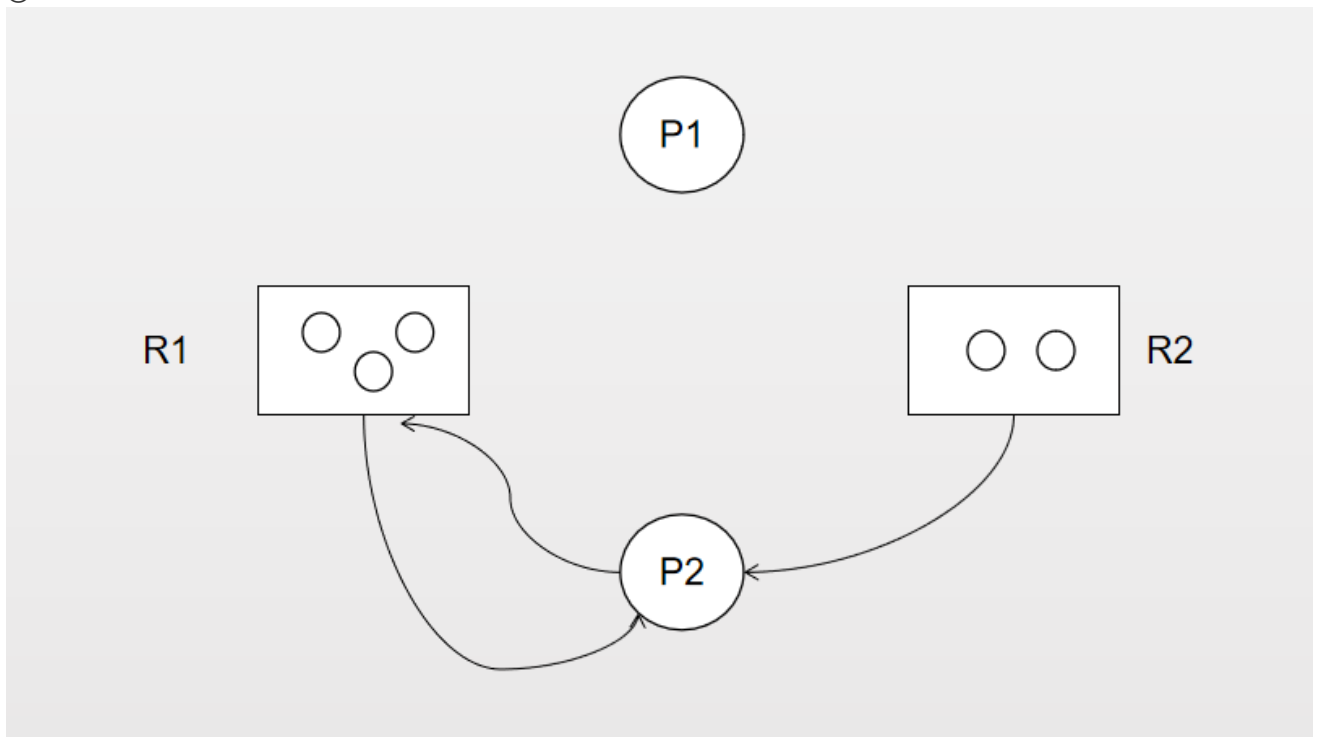




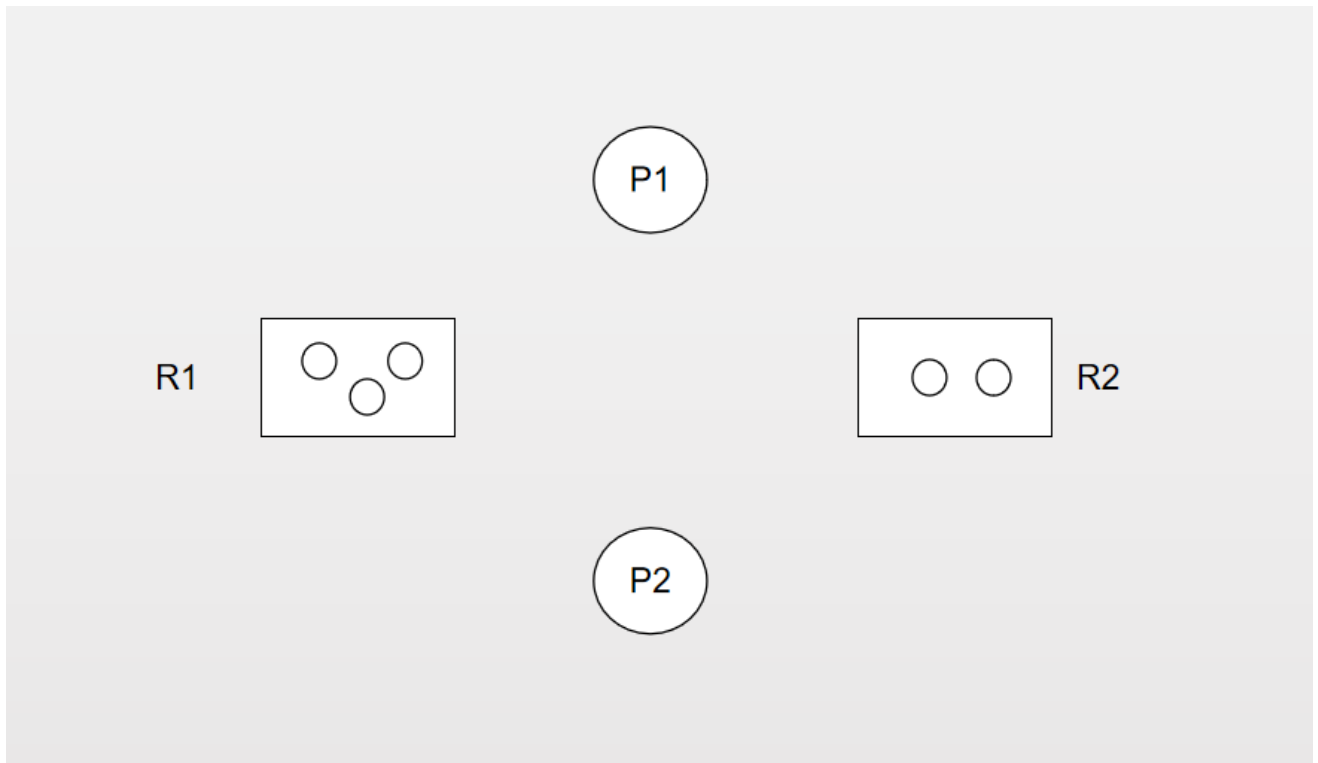
①寻找圆圈进程，判断申请边是否能够得到满足，满足则删除，不满足则找下一个进程。



②



③



- 根据资源转换图写出四个矩阵

$$Max = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix} \quad (5)$$

$$allocation = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \quad (5)$$

$$need = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (5)$$

$$available = [1 \quad 1] \quad (5)$$