



选9. 对于A选项 进程数为1, 显然不可能发生死锁

D. 易知 BCD选项均满足互斥和不可抢占以及部分持有并等待的条件而我们可以人为创造循环等待条件

对于B选项, 一共有两个进程, 两个资源但是由于两个进程的最大需求都是1, 则当进程1获得一个资源后调度至进程2, 进程2同样可以获得一个资源, 在任何情况下两个进程的需求都能得到满足, 故不会发生死锁

对于C选项, 一共有3个进程, 四个资源, 但是每个进程的最大需求都是2, 则当三个进程先分别拿到一个资源后无论接下来轮到哪个进程来, 其都能获得所需要的第二个资源然后释放掉然后其再轮到下一个进程获得其想要的进程, 则在任何情况下所有进程的需求都能得到满足, 故不会发生死锁

对于D选项, 一共有2个进程, 4个资源, 且每个进程的最大需求都是3, 则当每个进程先分别拿到两个资源时, 系统内已经没有可用资源, 无论接下来系统如何调度这两个进程, 其都不能再满足其最大需求, 故会出现死锁.

简4. (1) 先来先服务.

作业	提交	运行	开始	结束	周转时间T	带权周转时间w
1	8.0	1.0	8.0	9.0	1.0	1
2	8.5	0.5	9.0	9.5	1.0	2
3	9.0	0.2	9.5	9.7	0.7	3.5
4	9.1	0.1	9.7	9.8	0.7	7
平均					$\bar{T}=8.5$	$\bar{w}=3.375$

(2) 最短作业优先.

作业	提交时刻	运行时间	开始时间	结束时间	周转时间T	带权周转时间w
1	8.0	1.0	8.0	9.0	1.0	1
2	8.5	0.5	9.3	9.8	1.3	2.6
3	9.0	0.2	9.0	9.2	0.2	1
4	9.1	0.1	9.2	9.3	0.2	2
平均					$\bar{T}=0.675$	$\bar{w}=1.65$

$t=8.0$ 时, 只有一个作业1, 故令作业1上处理器运行.  $t=9.0$ 时, 等待队列中有作业2和3, 选择运行时间更短的作业3上处理器运行.  $t=9.2$ 时, 等待队列中有作业2和作业4, 选择运行时间更短的作业4上处理器运行.  $t=9.3$ 时, 等待队列中有作业2, 选择作业2上处理器运行. 得到的执行顺序为: 1, 3, 4, 2.





13) 最高响应比优先

$$\text{响应比} = \frac{\text{运行时间} + \text{等待时间}}{\text{等待时间}}$$

作业	提交时刻	运行时间	开始时刻	结束时刻	周转时间 $T$	带权周转时间 $\bar{w}$
1	8.0	1.0	8.0	9.0	1.0	1
2	8.5	0.5	9.0	9.5	1.0	2
3	9.0	0.2	9.6	9.8	0.8	4
4	9.1	0.1	9.5	9.6	0.5	5
平均					$\bar{T} = 0.825$	$\bar{w} = 3.0$

等待时间

0.5

0.5

0.4



$t=8.0$ 时,就绪队列中只有作业1.选择作业1上处理机运行

$t=9.0$ 时,就绪队列中有作业2和作业3.其响应比分别为2和1.选择作业2上处理机运行.

$t=9.5$ 时,就绪队列中有作业3和作业4,其响应比分别为3.5和4.选择作业4上处理机运行.

$t=9.6$ 时,就绪队列中只有作业3.选择作业3上处理机运行.

得到的执行顺序为 1, 2, 4, 3.

简5 (1) 是. 计算得此时 P1 进程的需求为  $\begin{bmatrix} 3, & 4, & 7 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 2, & 2, & 1 \\ 1, & 1, & 0 \end{bmatrix}$  剩余资源为 (2, 3, 3)

此时可以满足进程4和进程5. 我们先满足进程4. 再满足进程5.

此时的需求矩阵和剩余资源向量为  $\begin{bmatrix} 3, & 4, & 7 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 2, & 2, & 1 \\ 1, & 1, & 0 \end{bmatrix}$  (7, 4, 11)

此时 P1, 2, 3 均可以满足. 我们选择顺序为  $\begin{bmatrix} 3, & 4, & 7 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 2, & 2, & 1 \\ 1, & 1, & 0 \end{bmatrix}$  则安全序列为 4, 5, 1, 2, 3. 为安全状态.

(2) 假设能实施. 此时进程需求矩阵为  $\begin{bmatrix} 3, & 4, & 7 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 2, & 2, & 1 \\ 1, & 1, & 0 \end{bmatrix}$

由于 P1 资源只有3个. 无法满足 P2 的需求. 故不能实施.

(3). 此时剩余资源数可以满足 P4 的需求. 我们假设可以实施.

则此时的需求矩阵和剩余资源向量分别为  $\begin{bmatrix} 3, & 4, & 7 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 2, & 2, & 1 \\ 1, & 1, & 0 \end{bmatrix}$  (0, 3, 2) 此时剩余资源可以满足4进程

则此时的需求矩阵和剩余资源向量为  $\begin{bmatrix} 3, & 4, & 7 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 2, & 2, & 1 \\ 1, & 1, & 0 \end{bmatrix}$  (4, 3, 7) 此时可以满足进程2, 3, 5. 选择

进程5. 后面的进程同(1). 得到的安全序列为 4, 5, 1, 2, 3. 故可以实施

14) 此时剩余资源数可以满足 P4 的需求. 我们假设可以实施. 此时剩余资源数为 (0, 1, 2) 不能满足任何

进程需求. 则此时的需求矩阵和剩余资源向量分别为  $\begin{bmatrix} 3, & 2, & 5 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 2, & 2, & 1 \\ 1, & 1, & 0 \end{bmatrix}$  (2, 1, 3) 此时剩余资源可以满足进程5, 选择

此时的需求矩阵和剩余资源向量分别为  $\begin{bmatrix} 3, & 2, & 5 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 2, & 2, & 1 \\ 1, & 1, & 0 \end{bmatrix}$  (5, 2, 7) 此时剩余资源可以满足进程1, 3, 4.

我们选择 1, 3, 4 依次执行. 此时的需求和剩余资源为  $\begin{bmatrix} 0, & 7, & 0 \\ 1, & 3, & 4 \\ 0, & 0, & 6 \\ 0, & 0, & 0 \\ 0, & 0, & 0 \end{bmatrix}$  (13, 3, 18) 可以满足进程2.

得到安全序列为 5, 1, 3, 4, 2. 故可以实施.

实施 P4 的 (2, 0, 1) 后剩余资源为 (0, 1, 2) 此时满足不了任何进程的需求故不可实施.



扫描全能王 创建