

死锁习题分析：

1. 采用按序分配资源的策略可以预防死锁，这是利用了哪个条件不成立？

A. 互斥 B. 循环等待 C. 不可抢占 D. 占有并等待

【答案】B

【解析】按序分配资源的策略将所有的资源按类型进行分类，并赋予不同的序号。例如输入机的序号为 1，打印机序号为 2，磁盘机序号为 3 等。所有进程对资源的请求必须严格按资源序号递增的次序提出。这样在所形成的资源分配图中不可能再出现环路，因而“循环等待”条件不成立。在采用这种策略时总有一个进程占据了较高序号的资源，它继续请求的资源必然是空闲的，因而进程可以一直向前推进。

2. 存在一进程等待序列 $\{P_1, P_2, \dots, P_n\}$ ，其中 P_1 等待 P_2 所占有的某一资源， P_2 等待 P_3 所占有的资源，……而 P_n 等待 P_1 所占有的资源形成一个_____。

A. 进程顺序推进 B. 进程循环等待环
C. 资源有序分配 D. 资源强占

【答案】B

【解析】循环等待：若进程集合为 $\{P_0, P_1, \dots, P_n\}$ ，那么存在这样的关系， P_0 等待的资源被 P_1 所占有， P_1 等待的资源被 P_2 所占有， P_{n-1} 等待的资源被 P_n 所占有， P_n 等待的资源被 P_0 所占有。例如， n 个进程 P_1, P_2, \dots, P_n ， P_i ($i=1, \dots, n$) 因为申请不到资源 R_j ($j=1, \dots, m$) 而处于等待状态，而 R_j 又被 P_{i+1} ($i=1, \dots, n-1$) 占有， P_n 欲申请的资源被 P_1 占有，显然，此时这 n 个进程的等待状态永远不能结束。

3. 互斥条件是指_____。

A. 某资源在一段时间内只能由一个进程占有，不能同时被两个或两个以上的进程占有。
B. 一个进程在一段时间内只能占用一个资源。
C. 多个资源只能由一个进程占有。
D. 多个资源进程分配占有。

【答案】A

【解析】互斥条件：一个资源一次只能被一个进程所使用。如果进程 A 正在使用资源；如果另一进程 B 申请该资源，那么进程 B 的申请得不到满足，进程 B 只能等待，直至该资源被释放。

4. 某计算机系统中有 8 台打印机，由 K 个进程竞争使用，每个进程最多需要 3 台打印机。该系统可能发生死锁的 K 的最小值是

A. 2 B. 3 C. 4 D. 5

【答案】C

【解析】假设 $K=3$ 个，3 进程共享 8 台打印机，每个进程最多可以请求 3 台打印机，若 3 个进程都分别得到 2 台打印机，系统还剩下 2 台打印机，接下去无论哪个进程申请打印机，都可以得到满足，3 个进程都可以顺利执行完毕，这种情况下不会产生死锁。假设 $K=4$ ，4 个进程共享 8 台打印机，若 4 个进程都分别得到 2 台打印机，系统已经没有打印机了，接下去无论哪个进程申请打印机，都得不到满足，产生了相互等待，可能会发生死锁。因此答案是 C。

5. 设有一系统在某时刻的资源分配情况如下：

进程号 已分配资源 最大请求资源 剩余资源

	A B C D				A B C D				A B C D			
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

请问：

- (1) 系统中各进程尚需资源数各是多少？
- (2) 当前系统安全吗？
- (3) 如果此时进程 P1 提出资源请求 (0, 4, 2, 0)，系统能分配给它吗？

参考答案：

- (1) 尚需资源数矩阵如下：

$$\text{Need} = \text{Max} - \text{Allocation}$$

Need				
	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	1	0	0	2
P3	0	0	2	0
P4	0	6	4	2

- (2) 系统是安全的，因为可以找到一个安全序列：<P0, P2, P3, P4, P1>

- (3) 如 P1 申请(0,4,2,0)，则：

$$\text{Request1}(0,4,2,0) \leq \text{need1}(0,7,5,0)$$

$$\text{Request1}(0,4,2,0) \leq \text{available}(1,5,2,0)$$

新的状态为

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

该状态是安全的，存在安全序列如<P0, P2, P3, P4, P1>，所以可以分配资源给 P1。