作业二

**Part1 死锁**

1. 在操作系统中，死锁出现是指( A )

A.多个进程竞争资源出现了循环等待 B. 一个进程进入死循环

C. 进程释放资源 D. 多个进程竞争使用共享型的设备

2. 死锁的四个必要条件中，无法破坏的是( C )

A．非抢夺式分配 B．占有且等待资源

C．互斥使用资源 D．环路等待资源

3. 若系统 S1采用死锁避免方法，S2采用死锁检测方法。下列叙述中，正确的是( B )

I. S1会限制用户申请资源的顺序，而 S2 不会

II. S1需要进程运行所需的资源总量信息，而 S2不需要

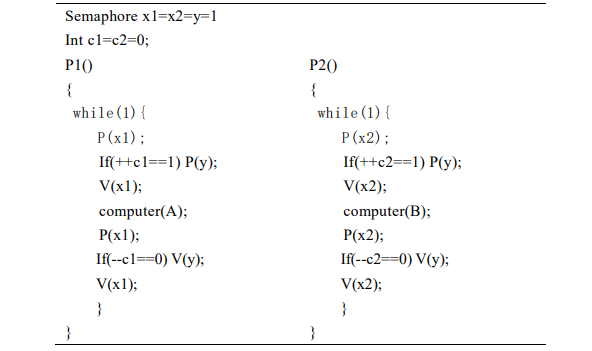
III. S1不会给可能导致死锁的进程分配资源，而 S2会

A．仅 I、II B．仅 II、III C．仅 I、III D．I、II、III

4、某系统有n台互斥使用的同类设备，三个并发进程分别需要3，4，5台设备，可确保系统不发生死锁的设备数n最小为( B )

A. 9 B. 10 C. 11 D. 12

5. 下面是一个并发进程的程序代码，正确的是( B )



A. 进程不会死锁，也不会“饥饿” B. 进程不会死锁，但是会“饥饿”

C. 进程会死锁，但是不会“饥饿” D. 进程会死锁，也会“饥饿”

6. 设系统中有如下解决死锁的方法：

1）银行家算法

2）检测死锁，终止处于死锁状态的进程，释放该进程占有的资源

3）资源预分配，即进程运行前一次性申请完所需要的全部资源

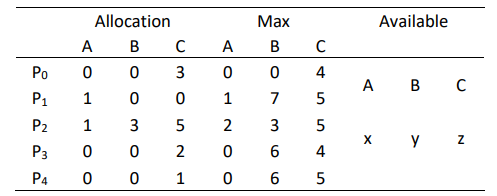
简述哪种办法允许最大的并发性，即哪种办法允许更多的进程无等待地向前推进。请按“并发性”从小到大对上述三种办法排序。

（1）银行家算法属于死锁避免，（2）检测死锁属于死锁检查，（3）资源预分配属于死锁预防

并发性大小为：死锁检查 > 死锁避免 > 死锁预防

故（1）（2）（3）排序为：（3）< （1） < （2）

7. 假设具有 5 个进程的进程集合 P={P0, P1, P2, P3, P4}，系统中有三类资源 A、B、C，假设在某时刻有如下状态，见下表。



请问当 x,y,z 取下列哪些值时，系统是处于安全状态的？给出计算过程。

I. 1,4,0 II. 0,6,2 III. 1,1,1 IV. 0,4,7

只有I是安全的，可以找到一个可行的序列为

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 进程 | Available | MAXNEED | Allocation | 可行性 |
| P2 | 1 4 0 | 1 0 0 | 1 3 5 | true |
| P1 | 2 7 5 | 0 7 5 | 1 0 0 | true |
| P0 | 3 7 5 | 0 0 1 | 0 0 3 | true |
| P3 | 3 7 8 | 0 6 2 | 0 0 2 | true |
| P4 | 3 7 10 | 0 6 4 | 0 0 1 | true |

II中最多满足P0、P3、P4，可用资源为0，6，8，此时P1和P2满足不了

III中B资源只有4个小于P1所需，从数量上可排除。

IV只能先满足P0，满足P0后可用资源为 0，4，10 满足不了任何进程。

**Part2 CPU调度**

1. 若某单处理器多进程系统中有多个就绪态进程，则下列关于处理机调度的叙述中，错误的是( D )。

A. 创建新进程后能进行处理机调度

B. 在进程结束时能进行处理机调度

C. 在系统调用完成并返回用户态时能进行处理机调度

D. 在进程处于临界区时不能进行处理机调度

1. P1、P2和P3是在某个系统中正在执行的三个进程，各进程的计算(CPU)时间和I/O时间比例如下表所示：

|  |  |  |
| --- | --- | --- |
| 进程 | 计算时间 | I/O 时间 |
| P1 | 50% | 50% |
| P2 | 20% | 80% |
| P3 | 30% | 70% |

为提高系统资源利用率，合理的进程优先级设置是( C )。

1. P1>P2>P3 B. P3>P2>P1 C. P2>P3>P1 D. P1>P3>P2
2. 下列选项中，降低进程优先级的合理时机是( B )。

A.进程刚完成I/O，进入就绪队列 B. 进程的时间片用完

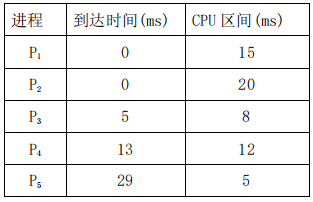
C.进程从就绪状态转为运行态 D. 进程长期处于就绪队列中

1. 陷阱指令（trap）可以使执行流程从用户态陷入内核，在用户进程使用陷阱（trap）执行调用内核函数的过程中，以下哪些步骤是由操作系统内核完成的( B )  
    ① 执行用户进程的main函数  
    ② 切换至内核模式  
    ③ 跳转至陷阱处理器（trap handler）  
    ④ 处理陷阱（handle trap）  
    ⑤ 执行系统调用

A.④⑤ B.③④⑤ C.②③④⑤ D.①⑤

1. 时钟中断（timer interrupt）是整个操作系统的脉搏，系统利用时钟中断维持系统时间、**保证进程共享CPU**、确定调度优先级。考虑如下场景：系统开机后，先执行进程A，接着遇到时钟中断，转为执行进程B，过程中需要执行以下操作  
    ① 从陷阱中返回（return from trap）  
    ② 初始化陷阱表（initialize trap table）  
    ③ 处理时钟中断（timer interrupt）  
    ④ 开启中断计时器（start interrupt timer）  
    ⑤ 执行进程A  
    ⑥ 执行进程B  
   操作的正确顺序是( D )  
   A.②④⑤③⑥① B. ⑤②④③⑥① C.④②⑤③①⑥ D.②④⑤③①⑥
2. 现有就绪进程 P1，P2，P3，P4，P5其到达时间和 CPU 区间如下表所示，0时

刻P1先到达，根据表格内容解决以下问题：



1) 若 Round-Robin (RR)调度方式，时间片为 10ms，画出 P1 到 P5的调度结果图

（维护一个就绪队列，轮转调度顺序按队列顺序依次进行调度）；

运行情况：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 调度时间 | 10ms | 10ms | 8ms | 5ms | 10ms | 10ms | 5ms | 2ms |
| 当前队列 | P2 P3 P1 | P3 P1 P4 P2 | P1 P4 P2 | P4 P2 P5 | P2 P5 P4 | P5 P4 | P4 | 空 |
| 执行进程 | P1 | P2 | P3 | P1 | P4 | P2 | P5 | P4 |

2) 若执行抢占式的 Shortest Job First (STCF)调度方式，画出 P1到 P5 的调度结果图；

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 运行时间 | 5ms | 8ms | 10ms | 6ms | 5ms | 6ms | 20ms |
| 执行进程 | P1 | P3 | P1 | P4 | P5 | P4 | P2 |

3) 分别计算两种模式下的平均响应时间和平均周转时间(假设不计进程调度所用时间)。

RR：

平均响应：（10+15+20+24） / 5 = 13.8ms

平均周转：（33 + 53 + 23 + 47 + 29） / 5 = 37ms

STCF

平均响应：（0 + 40 + 0 + 10 + 0） / 5 = 10ms

平均周转：（23 + 8 + 27 + 5 + 60） / 5 = 24.6ms