

1. 有五个进程P1、P2、P3、P4、P5，它们同时依次进入就绪队列，它们的优先数和需要的处理器时间如下表

进程	处理器时间	优先级（数小优先级高）
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

忽略进行调度等所花费的时间，回答下列问题：

- (1) 写出采用“先来先服务”、“短作业（进程）优先”、“非抢占式的优先数”和“轮转法”等调度算法，进程执行的次序。（其中轮转法的时间片为2）

先来先服务： P1, P2, P3, P4, P5

短作业优先： P2, P4, P3, P5, P1

非抢占式的优先数： P2, P5, P1, P3, P4

轮转法： P1, P2, P3, P4, P5, P1, P5, P1, P5, P1, P1

- (2) 分别计算上述算法中各进程的周转时间和等待时间，以及平均周转时间。

	周转时间	等待时间	平均周转时间
先来先服务	P1:10 P2:11 P3:13 P4:14 P5:19	P1:0 P2:10 P3:11 P4:13 P5:14	13.4
短作业优先	P1:19 P2:1 P3:4 P4:2 P5:9	P1:9 P2:0 P3:2 P4:1 P5:4	7
非抢占式的优先数	P1:16 P2:1 P3:18 P4:19	P1:6 P2:0 P3:16 P4:18	12

	P5:6	P5:1	
轮转法	P1:19 P2:3 P3:5 P4:6 P5:17	P1:9 P2:2 P3:3 P4:5 P5:10	10

2. 假设某系统使用时间片轮转调度算法进行CPU调度，时间片大小为5ms，系统共10个进程，初始时均处于就绪队列，执行结束前仅处于执行态或就绪态。若队尾的进程P所需的CPU时间为25ms，不考虑系统开销，则进程P的周转时间为？

周转时间为：  $25\text{ms} * 10 = 250\text{ms}$

3. 某个进程调度程序采用基于优先数(priority)的调度策略，即选择优先数最小的进程运行，运行创建时由用户指定一个nice作为静态优先数。为了动态调整优先数，引入运行时间cpuTime和等待时间waitTime，初值为0。进程处于执行态时，cpuTime定时加1，且waitTime置0；进程处于就绪态时，cpuTime置0，waitTime定时加1。

- (1) 若调度程序只将nice的值作为进程优先数，即priority = nice，则可能出现饥饿现象。为什么？

答：对于priority非常大的进程来说，如果一直有优先级很高的进程进来的话，该进程长时间得不到调度，会出现饥饿现象

- (2) 使用nice, cpuTime, waitTime设计一种动态优先数计算方法，以避免产生饥饿现象，并说明waitTime的作用

答：我们规定一种动态优先级  $\text{priority} = \text{nice} + \text{cpuTime} - \text{waitTime}$  的计算方法

该方法可以抑制长时间霸占CPU，同时也可以避免饥饿，此处waitTime的作用是让长时间的得不到调度的进程有机会逐步提高优先级，获得运行资源

4. (1) 有m个同类资源供n个进程共享，若每个进程最多申请k个资源 ( $k \geq 1$ )，采用银行家算法分配资源，为保证系统不发生死锁，则各进程的最大需求量之和为？并说明理由

答：为  $m + n - 1$ ，因为最坏情况是已经给所有进程分配了max\_i - 1个，此时只

需要再给一个进程分配一个就可以释放资源，故 $m - \sum(\max_i - 1) \geq 1$

(2) 有8台打印机，由K个进程竞争使用，每个进程最多使用3台打印机。求K的最小值，使系统可能发生死锁。

答：k为4得时候可能发生死锁，因为不会发生死锁得最大值为 $k + 8 - 1 \geq 3k$ ，即 $k \leq 3.5$

(3) 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。求n的最小值，使系统不发生死锁。

答：n最小为10，由于 $n + 3 - 1 \geq 12$ 可以得 $n \geq 10$

## 5. 什么是进程之间的同步关系？什么是进程之间的互斥关系？

进程同步：系统中各进程之间能有效地共享资源和相互合作，从而使程序得执行具有可再现性得过程称为进程同步

进程互斥：两个或两个以上的进程，不能同时进入关于同一组共享资源得临界区，否则可能发生与时间有关的错误

## 6. 假设具有5个进程的进程集合 $P = \{P_0, P_1, P_2, P_3, P_4\}$ ，系统中有三类资源 A,B,C，

假设在某时刻有如下状态：

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

(1) 根据上表内容，当前系统是否处于安全状态

答：是

(2) 若系统中的可利用资源 Available 为 (0,6,2)，系统是否安全？若系统处在安全状态，请给出安全序列；若系统处在非安全状态，简要说明原因。

答：安全

初始状态 (0, 6, 2)

P0可以满足需求 => (0, 6, 6)

P3可以满足需求 => (0, 12, 10)

P4可以满足需求 => (0, 18, 15)

P1可以满足需求 => (1, 25, 20)

P2可以满足需求 => (3, 28, 25)