

1. 有五个进程P1、P2、P3、P4、P5，它们同时依次进入就绪队列

1. 先来先服务

进程	进入时间	需要时间	开始时间	结束时间	等待时间	周转时间	带权周转时间
P1	0	10	0	10	0	10	1
P2	0	1	10	11	10	11	11
P3	0	2	11	13	11	13	6.5
P4	0	1	13	14	13	14	14
P5	0	5	14	19	14	19	3.8

进程执行次序：P1 → P2 → P3 → P4 → P5

平均周转时间： $\frac{10 + 11 + 13 + 14 + 19}{5} = 13.4$

2. 短进程优先

进程	进入时间	需要时间	开始时间	结束时间	等待时间	周转时间	带权周转时间
P1	0	10	9	19	9	19	1.9
P2	0	1	0	1	0	1	1
P3	0	2	2	4	2	4	2
P4	0	1	1	2	1	2	2
P5	0	5	4	9	4	9	1.8

进程执行次序：P2 → P4 → P3 → P5 → P1

平均周转时间： $\frac{19 + 1 + 4 + 2 + 9}{5} = 7$

3. 非抢占式的优先数

进程	进入时间	需要时间	优先数	开始时间	结束时间	等待时间	周转时间	带权周转时间
P1	0	10	3	6	16	6	16	3.2
P2	0	1	1	0	1	0	1	1
P3	0	2	3	16	18	16	18	9
P4	0	1	4	18	19	18	19	19
P5	0	5	2	1	6	1	6	1.2

进程执行次序：P2 → P5 → P1 → P3 → P4

平均周转时间： $\frac{16 + 1 + 18 + 19 + 6}{5} = 12$

4. 轮转法

进程	进入时间	需要时间	使用时间	结束时间	周转时间
P1	0	10	0-2, 8-10, 12-14, 15-19	19	19
P2	0	1	2-3	3	3
P3	0	2	3-5	5	5
P4	0	1	5-6	6	6
P5	0	5	6-8, 10-12, 14-15	15	15

进程执行次序：P1 → P2 → P3 → P4 → P5 → P1 → P5 → P1 → P5 → P1

平均周转时间： $\frac{19 + 3 + 5 + 6 + 15}{5} = 9.6$

2. 死锁产生的四个必要条件是什么？

1. **互斥条件**：指进程对所分配到的资源进行排它性使用，即在一段时间内某资源只由一个进程占用。如果此时还有其它进程请求资源，则请求者只能等待，直至占有资源的进程用毕释放。
 2. **请求和保持条件**：指进程已经保持至少一个资源，但又提出了新的资源请求，而该资源已被其它进程占有，此时请求进程阻塞，但又对自己已获得的其它资源保持不放。
 3. **不可剥夺条件**：指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。
 4. **环路等待条件**：指在发生死锁时，必然存在一个进程—资源的环形链，即进程集合{P0, P1, P2, ..., Pn}中的P0正在等待一个P1占用的资源；P1正在等待P2占用的资源；.....；Pn正在等待已被P0占用的资源。
3. 某系统中有n个进程和m台打印机，系统约定：打印机只能一台一台地申请、一台一台地释放，每个进程需要同时使用的打印机台数不超过m。如果n个进程同时需要使用打印机的总数小于m+n，试讨论，该系统可能发生死锁吗？并简述理由。

不可能发生死锁。证明如下：

\max_i 表示第*i*个进程的最大需求量；

$allocation_i$ 表示第*i*个进程的已分配资源量；

$need_i$ 表示第*i*个进程的还需要分配的量。

由题意可知： $\sum \max_i = \sum allocation_i + \sum need_i < m + n$

如果发生死锁，则表明所有*m*个资源都已经分配，即 $\sum allocation_i = m$ ，且进程无限等待资源。

上面两个式子相减，得到 $\sum need_i < n$ ，换言之，至少有一个进程没有申请资源，可以顺利执行完，并释放其所占资源。

与前面发生死锁的假设相矛盾。

4. 线程的基本概念是什么？

线程是一种“轻量级进程”，它是一个基本的CPU执行单元，也是程序执行流的最小单元，由线程ID、程序计数器、寄存器集合和堆栈组成。线程是进程中的一个实体，是被系统调度和分派的基本单位，线程自己不再拥有系统资源，只拥有一点儿在运行中必不可少的资源，但它可与同属于一个进程的其他线程共享所拥有的全部资源。

引入线程的好处是什么？

将资源与计算相分离，提高系统的并发效率。

5. 一个系统有4个进程和5个可分配资源，若保持该状态是安全状态，那么x的最小值是多少？

	Max	Allocation	Need	
进程A	11213	10211	1002	
进程B	22210	20110	2100	
进程C	21310	11010	10300	
进程D	11221	11110	111	

1. 假设 $x = 0$ ，Available = 12。由于所有四个进程的need都大于12，则全部等待，直接死锁。

2. 假设 $x = 1$ ，Available = 112。计算过程如下表所示。

	Work	Need	Allocation	Work + Allocation
进程D	112	111	11110	11222
进程A	11222	1002	10211	21433
进程B	21433	2100	20110	41543
进程C	41543	10300	11010	52553

根据上述计算流程，可以得到x的最小值是1。