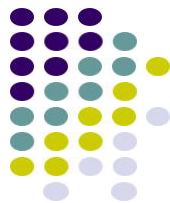


作业



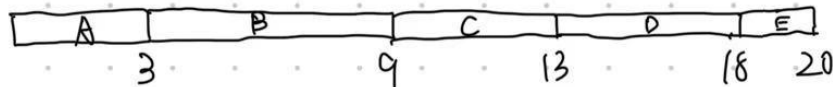
1、假设一个系统中有 5 个进程,它们的到达时间和服务时间如表 3-1 所示,忽略 I/O 以及其他开销时间,若分别按先来先服务(FCFS)、非抢占及抢占的短作业优先(SJF)、高响应比优先(HRRN)、时间片轮转(RR,时间片=1)、多级反馈队列调度算法(FB,第 i 级队列的时间片= 2^{i-1})进行 CPU 调度,请给出各进程的完成时间、周转时间、带权周转时间、平均周转时间和平均带权周转时间。

表 3-1 进程到达和需服务时间

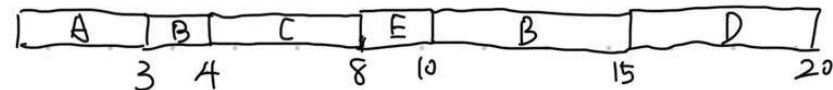
进程	到达时间	服务时间
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

未命名2

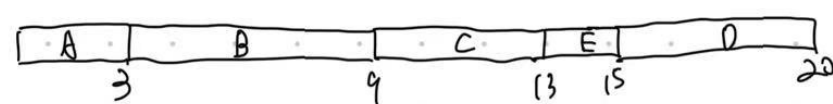
FCFS:



非抢占SJF:

抢占SJF:
(最短剩余时间)

HRRN (等待时间)



时间片轮转(RR):



FB: 一级队列: A(0~2) B(2~3) C(4~5) D(6~7) E(8~9)

二级队列: A(3~4) B(5~6) C(7~8) B(9~11) D(11~13) C(13~15) E(15~16)

三级队列: B(16~18) D(18~20)

FCFS:

进程	到达时间	服务时间	完成时间	周转时间	带权周转时间
A	0	3	3	3	1.00
B	2	6	9	7	1.17
C	4	4	13	9	2.25
D	6	5	18	12	2.40
E	8	2	20	12	6.00

平均周转时间: $(3+7+9+12+12)/5=8.6$

平均带权周转时间: $(1+1.17+2.25+2.4+6)/5=2.564$

非抢占式 SJF:

进程	到达时间	服务时间	完成时间	周转时间	带权周转时间
A	0	3	3	3	1.00
B	2	6	9	7	1.17
C	4	4	15	11	2.75
D	6	5	20	14	2.80
E	8	2	11	3	1.50

平均周转时间: $(3+7+11+14+3)/5=7.6$

平均带权周转时间: $(1+1.17+2.75+2.8+1.5)/5=1.844$

抢占式 SJF:

进程	到达时间	服务时间	完成时间	周转时间	带权周转时间
A	0	3	3	3	1.00
B	2	6	15	13	2.17
C	4	4	8	4	1.00
D	6	5	20	14	2.80
E	8	2	10	2	1.00

平均周转时间: $(3+13+4+14+2)/5=7.2$

平均带权周转时间: $(1.00+2.17+1.00+2.80+1.00)/5=1.594$

高响应比优先 (HRRN):

进程	到达时间	服务时间	完成时间	周转时间	带权周转时间
A	0	3	3	3	1.00
B	2	6	9	7	1.17
C	4	4	13	9	2.25
D	6	5	20	14	2.80
E	8	2	15	7	3.50

平均周转时间: $(3+7+9+7+14)/5=8$

平均带权周转时间: $(1.00+1.17+2.25+3.50+2.80)/5=2.144$

时间片轮转（RR）：

进程	到达时间	服务时间	完成时间	周转时间	带权周转时间
A	0	3	4	4	1.33
B	2	6	18	16	2.67
C	4	4	17	13	3.25
D	6	5	20	14	2.80
E	8	2	15	7	3.50

平均周转时间： $(4+16+13+14+7)/5=10.8$

平均带权周转时间： $(1.33+2.67+3.25+2.8+3.5)/5=2.71$

多级反馈队列调度（FB）：

进程	到达时间	服务时间	完成时间	周转时间	带权周转时间
A	0	3	4	4	1.33
B	2	6	18	16	2.67
C	4	4	15	11	2.75
D	6	5	20	14	2.80
E	8	2	16	8	4.00

平均周转时间： $(4+16+11+14+8)/5=10.6$

平均带权周转时间： $(1.33+2.67+2.75+2.80+4.00)/5=2.71$

作业



2、设在 t 时刻，系统中有 5 个进程，三种资源 A、B、C 的使用情况如下表所示，设系统可供使用的空闲资源数为 $Available=2, 2, 1$ 。若此时有以下三种不同的分配请求，请用银行家算法判断系统能否予以分配，并写出具体步骤。（注：以下各小题没有因果关系，每小题都以表中所示状态为当前状态）

- (1) P3 进程申请资源 A、B、C 分别为 2、0、0，能否分配，为什么
- (2) P1 进程申请资源 A、B、C 分别为 0、0、2，能否分配，为什么
- (3) P2 进程申请资源 A、B、C 分别为 1、1、1，能否分配，为什么

进程	Max			Need		
	A	B	C	A	B	C
P0	1	3	3	1	2	2
P1	3	2	2	2	2	1
P2	2	2	1	2	1	1
P3	2	0	2	1	0	1
P4	2	4	3	2	3	3

(1) 不能分配, 因为 $\text{Request}(A) > \text{Need}(A)$, P3 所需要的资源 A 的数量已经超过它所宣布的最大值

(2) 不能分配, 因为 $\text{Request}(C) > \text{Need}(C)$, P1 所需要的资源 C 的数量已经超过它所宣布的最大值

(3) 可以分配, 因为 $\text{Request}(A,B,C) < \text{Need}(A,B,C)$ 且 $\text{Request}(A,B,C) < \text{Available}(A,B,C)$, 分配资源后, $\text{Available}=1、1、0$, $\text{P2.Need}=1、0、0$, 还是能够分配给 P2, 使 P2 进程运行, 不会发生死锁。

安全序列为: P2 -> P1 -> P3 -> P0 -> P4