

5.6 轮转调度程序的一个变种是回归轮转 (regressive round-robin) 调度程序。这个调度程序为每个进程分配时间片和优先级。时间片的初值为 50ms。然而, 如果一个进程获得 CPU 并用完它的整个时间片 (不会因 I/O 而阻塞), 那么它的时间片会增加 10ms 并且它的优先级会提升。(进程的时间片可以增加到最多 100ms。) 如果一个进程在用完它的整个时间片之前阻塞, 那么它的时间片会降低 5ms 而它的优先级不变。回归轮转调度程序会偏爱哪类进程 (CPU 密集型的或 I/O 密集型的)? 请解释。

偏爱 CPU 密集型进程:

CPU 密集型进程不会因为 I/O 操作阻塞; 在回归轮转调度中, 可以用完整个时间片, 因此获得的时间片会逐步增加, 优先级也会提升, 可以进一步提升运行效率。

而 I/O 密集型进程通常在用完时间片之前就会因为 I/O 操作阻塞, 因此时间片会逐步减少, 优先级保持不变, 因此, 它们每次被调度时能使用的时间片会越来越短, 降低了执行效率。

5.7 假设有如下组进程，它们的 CPU 执行时间以毫秒来计算：

进程	执行时间	优先级
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

假设进程按 P_1 、 P_2 、 P_3 、 P_4 、 P_5 顺序在时刻 0 到达。

- 画出 4 个 Gantt 图，分别演示采用每种调度算法（FCFS、SJF、非抢占优先级（一个较大优先级数值意味着更高优先级）和 RR（时间片=2））的进程执行。
- 每个进程在 a 里的每种调度算法下的周转时间是多少？
- 每个进程在 a 里的每种调度算法下的等待时间是多少？
- 哪一种调度算法的平均等待时间（对所有进程）最小？

5.7

a. FCFS:

SJF:

非抢占优先级:

RR:

b.

进程	P_1	P_2	P_3	P_4	P_5
FCFS	2	3	11	15	20
SJF	3	1	20	7	12
非抢占优先级	15	20	8	19	13
RR	2	3	20	13	18

c.

进程	P_1	P_2	P_3	P_4	P_5
FCFS	0	2	3	11	15
SJF	1	0	12	3	7
非抢占优先级	13	19	0	15	8
RR	0	2	7	9	13

d. FCFS: $t = \frac{0+2+3+11+15}{5} = \frac{31}{5} = 6.2ms$ SJF: $t = \frac{1+0+12+3+7}{5} = \frac{23}{5} = 4.6ms$
 非抢占优先级: $t = \frac{13+19+0+15+8}{5} = 11ms$ RR: $t = \frac{0+2+7+9+13}{5} = \frac{31}{5} = 6.2ms$
 \therefore SJF 的平均等待时间最小

5.12 现有运行 10 个 I/O 密集型任务和 1 个 CPU 密集型任务的一个系统。假设 I/O 密集型任务每 1ms 的 CPU 计算就进行一次 I/O 操作，并且每个 I/O 操作需要 10ms 来完成。另假设上下文切换开销是 0.1ms，所有进程都是长时间运行的任务。请讨论在下列条件下轮转调度程序的 CPU 利用率：

- 时间片为 1ms
- 时间片为 10ms

5.12. a. \therefore 1ms 后，I/O 密集型任务刚好执行 I/O 操作，且 10ms 后它才再次上 CPU 运行

\therefore 每次进程运行只需要加上 0.1ms 的上下文切换开销

$$\therefore \text{CPU 利用率} = \frac{1}{1.1} = 90.91\%$$

b. 总时间 = 上下文切换开销 + I/O 任务运行时间 + CPU 任务运行时间

$$= 11 \times 0.1 + 10 + 10 \text{ms}$$

$$= 21.1 \text{ms}$$

$$\text{CPU 运行时间} = 20 \text{ms}$$

$$\therefore \text{CPU 利用率} = \frac{20}{21.1} = 94.79\%$$

29	19	89	1	2	9
05	21	11	8	1	2
14	5	05	1	8	1
13	14	8	05	21	1
18	13	05	8	1	1