

# 操作系统 (2024-2025)

## 作业 #3: 241118 进程调度

截止日期: 2024 年 11 月 27 日

张梓卫 (学号: 10235101526)

## 问题 1

### 1 5.6

轮转调度程序的一个变种是回归轮转 (regressiveround-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 IO 密集型的)? 请解释。

#### 解答

回归轮转调度程序会偏爱 CPU 密集型的进程, 这种调度器将有利于 CPU 受限的进程, 因为它们会获得更长的时间量以及每当它们消耗整个时间量时, 优先级就会提高。因为它们能够连续完成分配的时间片, 时间片和优先级会逐渐提高。此调度程序不会惩罚 I/O 限制进程, 因为它们在消耗全部时间之前可能会阻塞 I/O, 但它们的优先级保持不变。

### 2 5.7

假设有如下 1 组进程, 它们的 CPU 执行时间可以毫秒来计算:

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、非抢占优先级调度 (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?

#### 解答

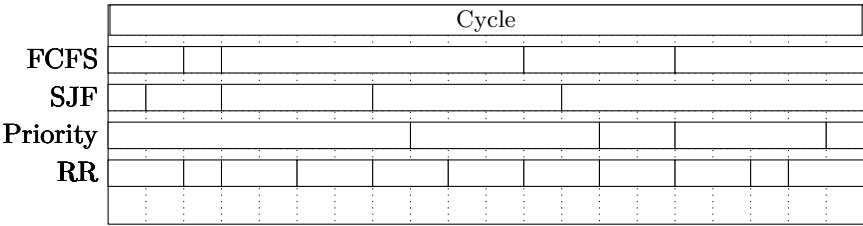


图 1: Gantt Chart

(b) 周转时间 = 完成时间 - 到达时间

	P1	P2	P3	P4	P5
FCFS	2ms	3ms	11ms	15ms	20ms
SJF	3ms	1ms	20ms	7ms	12ms
非抢占优先级	15ms	20ms	8ms	19ms	13ms
RR	2ms	3ms	20ms	13ms	18ms

(c) 等待时间 = 周转时间 - 执行时间

	P1	P2	P3	P4	P5
FCFS	0ms	2ms	3ms	11ms	15ms
SJF	1ms	0ms	12ms	3ms	7ms
非抢占优先级	13ms	19ms	0ms	15ms	8ms
RR	0ms	2ms	12ms	9ms	13ms

调度算法	平均等待时间
FCFS	6.2ms
SJF	4.6ms
非抢占优先级	11.0ms
RR	7.2ms

表 1: 各调度算法的平均等待时间

(d) Shortest Job First

- Shortest Job First (SJF) minimizes the average waiting time.

### 3 5.12

现有运行 10 个 IO 密集型任务和 1 个 CPU 密集型任务的一个系统。假设 IO 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:

- (a) 时间片为 1ms
- (b) 时间片为 10ms

#### 解答

##### 3.1 a

时间量为 1 毫秒: 无论调度哪个进程, 调度器每次切换上下文, 都会产生 0.1 毫秒的上下文切换成本。这导致 CPU 利用率为  $\frac{1ms}{1ms+0.1ms} \times 100\% \approx 90.91\%$ .

##### 3.2 b

时间量为 10 毫秒: I/O 受限的任务在仅使用 1 毫秒后就会发生上下文切换毫秒的时间量。因此, 循环完成所有流程所需的时间为  $10 \times 1.1 + 10.1$  (因为每个 I/O 绑定任务执行 1 毫秒, 然后引发上下文切换任务, 而 CPU 限制的任务在引发上下文切换之前执行 10 毫秒)。CPU 利用率为  $\frac{20ms}{21.1ms} \times 100\% \approx 94.79\%$ .