

操作系统原理第五章作业

姓名：马福泉 学号：23336179 截止日期：2025年4月15日

完成日期：2025年4月8日

Question 1：为什么对调度程序而言，区分 CPU 约束程序和 I/O 约束程序很重要？

Answer 1：

(1) CPU 约束程序：这类程序主要进行计算密集型任务，需要长时间占用 CPU，但对 I/O 资源需求较少。

(2) I/O 约束程序：这类程序需要频繁进行 I/O 操作，每次占用 CPU 的时间很短，但 I/O 等待时间较长。

(3) 通过区分程序类型，调度程序可以合理分配 CPU 时间，确保每个程序都能在合理的时间内获得 CPU 资源：优先调度 I/O 约束程序，可以减少用户等待时间，提高系统的响应速度。反之，CPU 约束程序占用过多 CPU 时间，可能会导致 I/O 约束程序等待时间过长，降低系统的整体响应性能。

Question 2：讨论下列几对调度标准如何在一定设置中冲突：

- a.CPU 利用率和响应时间
- b.平均周转时间(turnaroundtime)和最大等待时间
- c.IO 设备利用率和 CPU 利用率

Answer 2：

(a) 高 CPU 利用率（如长任务优先）会让 CPU 保持忙碌，这会导致交互式任务等待，增加响应时间；而快速响应（如时间片轮转）需要频繁切换进程，降低 CPU 利用率。

(b) 降低平均周转时间（如短作业优先），短作业快速完成，但长作业等待时间过长，最大等待时间增加；控制最大等待时间（如公平调度），作业执行顺序均匀，但短作业无法快速完成，平均周转时间上升。

(c) 提高 CPU 利用率，计算密集型任务（CPU 约束程序）优先，IO 密集型任务（I/O 约束程序）调度少，IO 设备闲置；提高 IO 设备利用率，优先调度 IO 密集型任务，CPU 等待 IO 完成，利用率下降。

Question 3: 考虑用于预测下一个 CPU 区间长度的指数平均公式。将下面的值赋给算法中的参数的含义是什么？

- a. $\alpha=0$ 且 $t_0=100 \text{ ms}$
- b. $\alpha=0.99$ $t_0=10 \text{ ms}$

Answer 3:

$$T_{n+1} = \alpha t_n + (1-\alpha) T_n$$

当 $\alpha=0$ 时 $T_0=100 \text{ ms}$

即 $T_{n+1} = T_n = \dots = T_0 = 100 \text{ ms}$

即预测值 T_{n+1} 完全忽略观测值 t_n , 始终等于历史预测值 $T_0=100 \text{ ms}$

当 $\alpha=0.99$ 时 $T_0=10 \text{ ms}$

$$T_{n+1} = 0.99 t_n + 0.01 T_n$$

即预测值 T_{n+1} 几乎完全由观测值 t_n 决定, 而历史预测值权重很低 (仅为 1%)

即预测值对最近观测值很敏感, 会随着其变化而快速变化

Question 4:

5.4 考虑下面一组进程，进程占用的 CPU 区间长度以毫秒来计算：

进程	区间时间	优先级
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

假设在 0 时刻进程以 P_1, P_2, P_3, P_4, P_5 的顺序到达。

- a. 画出 4 个 Gantt 图分别演示使用 FCFS、SJF、非抢占优先级（数字越小代表优先级越高）和 RR（时间片=1）算法调度时进程的执行过程。
- b. 每个进程在每种调度算法下的周转时间是多少？
- c. 每个进程在每种调度算法下的等待时间是多少？
- d. 哪一种调度算法的平均等待时间最小（对所有的进程）？

Answer 4:

(a) FCFS

	P1	P2	P3	P4	P5
D	10	11	13	14	19

SJF

	P2	P4	P3	P5	P1
D	0	1	2	4	9

非抢占式优先级

	P2	P5	P1	P3	P4
D	0	1	b	16	18

RR

	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5
D	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

(b) 周转时间

	FCFS	SJF	Priority	RR
P1	10	19	16	19
P2	11	1	1	2
P3	13	4	18	7
P4	10	2	19	4
P5	19	9	6	14

(c) 等待时间

	FCFS	SJF	Priority	RR
P1	0	9	b	9
P2	10	0	0	1
P3	11	2	16	5
P4	13	1	18	3
P5	10	4	1	9

(d) 平均等待时间

$$FCFS: (0+11+13+14)/5 = 9.6$$

$$SJF: (9+2+1+4)/5 = 3.2$$

$$Priority: (6+16+18+1)/5 = 8.2$$

$$RR: (9+1+5+3+9) = 5.4$$

SJF 平均等待时间最小

Question 5: 下面哪种调度算法能导致饥饿?

- a. 先到先服务
- b. 最短作业优先
- c. 轮转法
- d. 优先级

Answer 5: b、d

(1) 会导致饥饿

最短作业优先: 该算法优先执行运行时间最短的作业。如果系统持续有短作业到达, 长作业可能会被无限期推迟, 导致长作业饥饿。

优先级调度：低优先级的进程可能永远得不到执行机会，尤其是静态优先级时，
高优先级进程持续到达会导致低优先级进程饥饿。

(2) 不会导致饥饿

先到先服务：每个作业最终都会按到达顺序执行。

轮转法：通过时间片轮转，所有进程都能公平获得 CPU 时间，一般不会饥饿。