

11/8 20:29 電資三 黃曉筠

4.8

1. 一個超快的任務 e.g. `print("Hello world")`
2. 需要按順序執行的東西 e.g. 先付款後才能發貨

4.14

$$\text{Speedup} = \frac{1}{(1-p) + \frac{p}{n}} \quad (p = \text{parallel}, n = \text{core nums})$$

1-a

$$\frac{1}{(1-0.4) + \frac{0.4}{8}} \approx 1.538$$

1-b

$$\frac{1}{(1-0.4) + \frac{0.4}{16}} \approx 1.6$$

2-a

$$\frac{1}{(1-0.67) + \frac{0.67}{8}} \approx 1.504$$

2-b

$$\frac{1}{(1-0.67) + \frac{0.67}{4}} \approx 2.010$$

3-a

$$\frac{1}{(1-0.9) + \frac{0.9}{4}} \approx 3.077$$

3-b

$$\frac{1}{(1-0.9) + \frac{0.9}{8}} \approx 4.706$$

4.20

a. kernel threads < process cores

CPU 閒置, Scheduling Bottleneck (有工作 but 排不出去)

b. kernel threads = process cores

CPU 滿載, User-level scheduling overhead 問題依舊在

c. kernel threads > process cores

Excessive CPU Context Switching (切換時間太多效率變差), cache pollution

5.12

a.

CPU utilization: 讓 CPU 隨時有工作做 (使用率接近 100%)

Response time: 使用者請求 → 回應的時間 (越短越好)

⇒ 高 CPU utilization = 優先安排長時間運算的作業

= interactive 請求回應時間變長

5.12

b.

Average turnaround time : 所有作業提交到完成的平均時間 (越短越好)

Maximum waiting time : 所有作業中等待最久的時間 (越短越好)

\Rightarrow 高 Average turnaround time = shortest job first

- = 長作業一直被短作業擋隊
- = 長作業等待時間超長

c. I/O device utilization : 讓 I/O 設備不閒置

CPU utilization : 讓 CPU 不閒置

 \Rightarrow I/O bound = CPU idle

CPU bound = I/O devices idle

5.17

a. b. c. in P.3

d.

$$FCFS = \frac{0+5+8+9+16}{5} = 7.6$$

$$SJF = \frac{8+1+0+13+4}{5} = 5.2 \#$$

$$Priority = \frac{0+17+9+10+5}{5} = 8.2$$

$$RR : \frac{12+9+4+13+12}{5} = 10$$

5.22

$$a. (1 + 0.1)_{(tasks)} \times 11_{(tasks)} = 12.1 ms$$

$$\frac{11 ms (CPU)}{12.1 (total)} \approx 90.91\% \#$$

$$b. 10 (I/O bound) + 10 (CPU bound) + 11 \times 0.1 (switch) = 21.1 ms$$

$$\frac{20}{21.1} \approx 94.79\% \#$$

5.25

a. FCFS X

可能造成 convoy effect (一大群短程序被長程序擋住)

b. RR Δ

短程序可能在幾次 quantum 完成

c. Multilevel Feedback Queues 0

短程序留在優先 queue 上, 幾乎立即被完成

6.7

(a)

top() 指標混亂

push() 和 pop() 都會對 top 進行讀取/修改

2 執行緒同時操作 \Rightarrow 資料混亂

(b) lock or mutex / atomic operations

6.15

1. 使用者無權 disable interrupts

(防止惡意程式影響整個系統)

2. 模擬 disable interrupts 無法保證 Atomic

(排程器優先, 可能中途會被 Context-switch)

3. 影響整體系統響應能力

4. 影響不到 multi-thread 的切換

6.18

1. Waiting Queue : 將無法獲得 lock 的程式加入其中

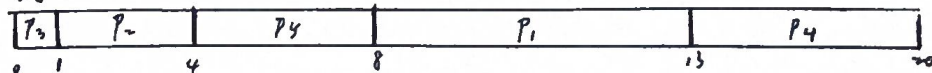
2. 釋放鎖後從 waiting queue 中選一個 wake up

5.17

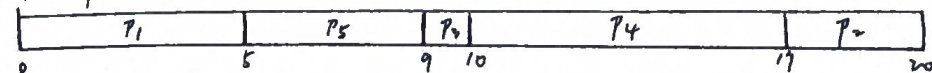
a. FCFS



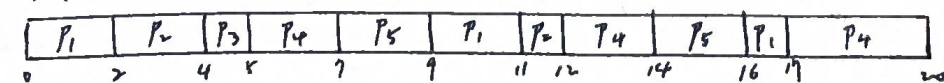
SJF



Priority



RR



b.

Process	FCFS	SJF	Priority	RR	Process	FCFS	SJF	Priority	RR
---------	------	-----	----------	----	---------	------	-----	----------	----

P1	5	13	5	17	P1	0	8	0	12	-5
P2	8	4	20	12	P2	5	1	17	9	-3
P3	9	1	10	5	P3	8	0	9	4	-1
P4	16	20	17	20	P4	9	13	10	13	-7
P5	20	8	9	16	P5	16	4	5	12	-4