Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

Each answer shows the steps, then the **result**.

2.1

P1 0, P2 5

P3 8

average $13/3 \approx$ **4.33** ms

2.2

order P3, P2, P1: waiting 0, 1, 4

average $5/3 \approx$ **1.67** ms

2.3

P1 0–2, P2 2–4, P3 4–5

P1 5–7, P2 7–8, P1 8–9

completions: P3 5, P2 8, P1 9

2.4

a low-priority job never runs, because higher-priority jobs keep arriving

aging raises a job's priority the longer it waits, so it runs in the end

3.1

(a) P1 0–2; P2 2–4 (P2's 4 < P1's 5 left); P3 4–5; P2 5–7; P4 7–11; P1 11–16

(b) turnaround: P1 16, P2 5, P3 1, P4 6; average $28/4 =$ **7** ms

(c) waiting: 9, 1, 0, 2; average $12/4 =$ **3** ms

(d) it needs the length of each next burst, which is not known; it must be estimated

3.2

(a) $9 \times (20 + 1) =$ **189** ms; overhead $1/21 \approx$ **4.8**%

(b) $9 \times 101 =$ **909** ms; overhead $1/101 \approx$ **1**%

(c) $q = 20$ ms: a wait of almost 1 s feels slow, and 4.8% overhead is acceptable

3.3

(a) $\tau_1 = 0.5 \times 6 + 0.5 \times 10 =$ **8**; $\tau_2 = 0.5 \times 4 + 0.5 \times 8 =$ **6**; $\tau_3 = 6$, $\tau_4 = 5$

(b) $\alpha = 1$: only the last burst counts; $\alpha = 0$: the estimate never changes from τ_0

(c) several queues, the top with a short quantum and highest priority; a job that uses its whole quantum moves down, so interactive jobs stay high and batch jobs sink to long-quantum queues

exercises.lol exercises.com.cn

(d) count the total processor time a job uses at a level, and move it down when that passes the limit, however the time was split