운영체제 (062 분반)

Ch 05. CPU Scheduling

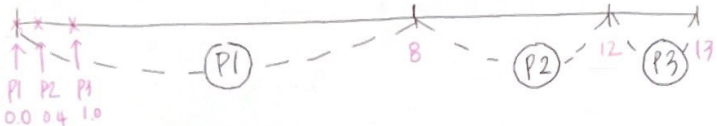
Assignment

학과 : 정보컴퓨터공학부
학번 : 2019244437
이름 : 김윤하
제출일 : 2023.04.16

5.3. FCFS : First Come First Served (선입선출) 알고리즘이다.

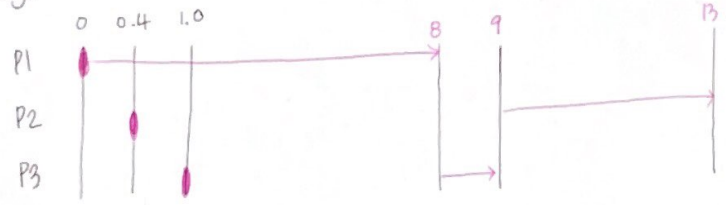
	도착 시간	실행 시간
P1	0.0	8
P2	0.4	4
P3	1.0	1.

a. FCFS
Average turnaround time ; 총 처리 시간 (P 시작 → 끝 까지의 시간 !)



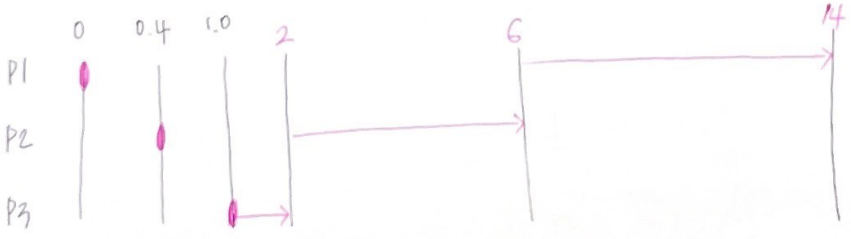
$$\therefore \frac{(8-0)+(12-0.4)+(13-1.0)}{3} = \frac{8+11.6+12}{3} = 10.53$$

b. SJF : CPU 점유 시간이 짧은 것부터 보내주는 방법.
Average turnaround time



$$\therefore \frac{(8-0)+(9-1)+(13-0.4)}{3} = \frac{8+8+12.6}{3} = 9.53$$

c.



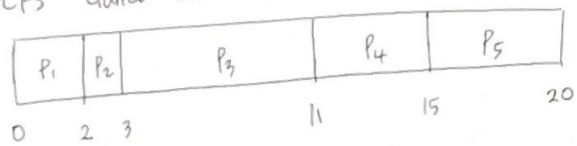
$$\therefore \frac{(2-1)+(6-0.4)+(14-0)}{3} = \frac{1+5.6+14}{3} = 6.87$$

5.4.

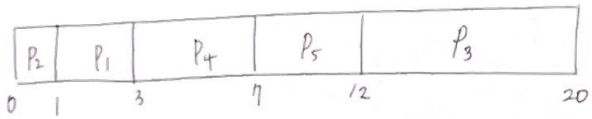
" P1 → P2 → P3 → P4 → P5 "

	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

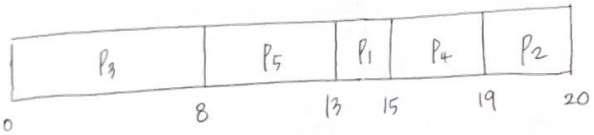
2. ① FCFS Gantt chart



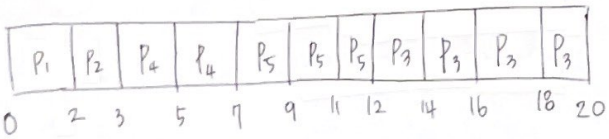
② SJF Gantt chart



③ non-preemptive priority Gantt chart



④ RR (quantum = 2) Gantt chart



b. turnaround time of each process

① FCFS

P_1 : 2 , P_2 : 3 , P_3 : 11 , P_4 : 15 , P_5 : 20

② SJF

P_1 : 3 , P_2 : 1 , P_3 : 20 , P_4 : 7 , P_5 : 12

③ non-preemptive priority

P_1 : 15 , P_2 : 20 , P_3 : 8 , P_4 : 19 , P_5 : 13

④ RR(quantum = 2)

P_1 : 2 , P_2 : 3 , P_3 : 20 , P_4 : 7 , P_5 : 12

c. waiting time of each process

① FCFS

P_1 : 0 , P_2 : 2 , P_3 : 3 , P_4 : 11 , P_5 : 15

② SJF

P_1 : 1 , P_2 : 0 , P_3 : 12 , P_4 : 3 , P_5 : 7

③ non-preemptive priority

P_1 : 13 , P_2 : 19 , P_3 : 0 , P_4 : 15 , P_5 : 8

④ RR(quantum = 2)

P_1 : 0 , P_2 : 2 , P_3 : 12 , P_4 : 3 , P_5 : 7

d. minimum Avg. waiting time ?

① $\frac{31}{5}$ ② $\frac{23}{5}$ ③ $\frac{55}{5}$ ④ $\frac{24}{5}$

∴ ② SJF가 가장 짧다.

5.7.

a. Priority and SJF

SJF는 실행 시간 (Burst time) 기반으로 우선 순위를 계산하며,

Priority 알고리즘은 SJF 알고리즘의 하위 집합이다.

b. Multilevel feedback queues and FCFS

서로 관련이 없는 알고리즘이다.

multilevel feedback queues는 다중 큐를 사용하고, process queue를 간에 이동하지만,
(우선순위 기준 동작)

FCFS는 먼저 도착한 process를 먼저 실행한다.

c. Priority and FCFS

서로 관련이 없는 알고리즘이다.

Priority → 우선 순위에 따라 process 선택해 실행.

FCFS → 그냥 도착 순서대로 process 실행.

d. RR and SJF

서로 관련이 없는 알고리즘이다.

RR는 time quantum을 할당해 그 만큼 내에서 프로세스를 실행한다.

(큰 틀은 큰 순서대로 동작.)

SJF는 시간이 짧은 것부터 먼저 실행된다.

RR는 시간이 긴 process 더라도, 차례가 오면 반드시 실행된다.

5.19.

nice value는 낮은 값일수록 높은 우선순위를 가진다.

만약 모든 사용자가 nice value < 0 을 할당할 수 있다면,

악의적인 사용자가 시스템의 CPU 성능을 오남용하는 등의 부정한 작업을 수행할 수 있다.

반면, nice value ≥ 0 은 CPU 시간 할당에 영향을 미치지만,

시스템 성능에 큰 영향을 주지 않는다.

5.22.

I/O → 10개 (10ms), CPU bound → 1개.
context-switch overhead = 0.1 ms

a. time quantum = 1 ms

→ I/O 대기 시간 매우 짧다. (CPU bound 작업 대기가 없음)

context-switch

$$\frac{\text{10번의 I/O process + 1번 CPU process}}{\text{10번 I/O process + 1번 CPU process + 10번 context-switch}} = \frac{10ms + 1ms}{10ms + 1ms + 1ms}$$

$$= \frac{11}{12} ms \cong \boxed{91.67\ \%}$$

b. time quantum = 10 ms

quantum 안에 완전히 가능.

$$\frac{\text{10번 I/O process + 1번 CPU process}}{\text{10번 I/O process + 1번 CPU process + 10번 context-switch}} = \frac{10ms + 10ms}{10ms + 10ms + 1ms}$$

$$= \frac{20}{21} ms \cong \boxed{95\ \%.}$$