

운영체제 과제 [6장 연습문제 풀이]

201810912 컴퓨터과학과 김연출

6.1 A CPU-scheduling algorithm determines an order for the execution of its scheduled processes. Given n processes to be scheduled on one processor, how many different schedules are possible? Give a formula in terms of n .

- $n!$ (n factorial)

6.9 The traditional UNIX scheduler enforces an inverse relationship between priority numbers and priorities: the higher the number, the lower the priority. The scheduler recalculates process priorities once per second using the following function: $\text{Priority} = (\text{recent CPU usage} / 2) + \text{base}$ where $\text{base} = 60$ and recent CPU usage refers to a value indicating how often a process has used the CPU since priorities were last recalculated. Assume that recent CPU usage is 40 for process P1, 18 for process P2, and 10 for process P3. What will be the new priorities for these three processes when priorities are recalculated? Based on this information, does the traditional UNIX scheduler raise or lower the relative priority of a CPU-bound process?

- $\text{base} = 60$, recent CPU usage $p1 = 40$, $p2 = 18$, $p3 = 10$ 이므로

Recalculated priority: $p1 = (40/2)+60$, $p2=(18/2)+60$, $p3=(10/2)+60$

$P1 = 80$, $p2 = 69$, $p3 = 65$

우선순위는 $p3 > p2 > p1$ 이다.

스케줄러는 cpu 사용시간이 길수록 우선순위를 낮게 설정하므로 CPU-bound-processes의 상대적 우선 순위를 낮춥니다 (lower)

6.16 Consider the following set of processes, with the length of the CPU burst given in milliseconds: Process Burst Time Priority P1 2 2 P2 1 1 P3 8 4 P4 4 2 P5 5 3 The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

<u>Process</u>	<u>Burst Time</u>	<u>Priority</u>
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

- a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).

-FCFS

P1	P2	P3	P4	P5	
0	2	3	11	15	20

-SJF

P2	P1	P4	P5	P3	
0	1	3	7	12	20

-nonpreemptive priority

P3	P5	P1	P4	P2	
0	8	13	15	19	20

- RR

P1	P2	P3	P4	P5	P3	P4	P5	P3	P5	P3	
0	2	3	5	7	9	11	13	15	17	18	20

- b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

	FCFS	SJF	NON-PREEM.	RR
P1	2	3	15	2
P2	3	1	20	3
P3	11	20	8	20
P4	15	7	19	13
P5	20	12	13	18

- c. What is the waiting time of each process for each of these scheduling algorithm?

	FCFS	SJF	NON-PREEM.	RR
P1	0	1	13	0
P2	2	0	19	2
P3	3	12	0	12
P4	11	3	15	9
P5	15	7	8	13

d. Which of the algorithms results in the minimum average waiting time (over all processes)?

FCFS = 6.2

SJF = 4.6

Non- preem. = 11.4

RR = 7.2

➔ SJF is minimum average waiting time