

OS - 4. CPU 스케줄링 알고리즘(1)

CPU 스케줄링

- 현재 CPU에서 실행중인 프로세스가 끝나고 나면 어느 프로세스를 실행시킬 지 결정하는 것

🐇 Preemptive vs Non-preemptive

🥕 Preemptive(선점) : 미리 점유

--> CPU가 어떤 프로세스를 실행하고 있을 때 아직 CPU가 끝나지도 않았고, IO를 만난 것도 아닌데 강제로 쫓아내고 새로운 프로세스가 들어갈 수 있도록 허용하는 스케줄링

--> 예) 병원에서 환자가 진료받고 있는데 응급 환자가 와서 진료 중이던 환자를 내보내고 응급환자를 먼저 받도록 허용하는 것

🥕 Non-preemptive(비선점) : 미리 점유하지 않음

--> 이미 CPU가 실행중이면, CPU가 끝나거나 IO를 만나기 전까지는 절대 스케줄링이 일어나지 않는다.

--> 예) 병원에서 환자가 진료를 받고 있는 동안에는 중간에 환자를 내보내고 다른 사람이 진료를 받을 수 없다.

🐇 Scheduling criteria(스케줄링 비교척도 --> 이 스케줄링으로 하니 이런 점이 좋고, 저 스케줄링으로 하니 저런 점이 좋다는 걸 비교하는 척도!)

- CPU Utilization(CPU 이용률) : CPU를 얼마나 활용해 작업을 빨리 끝내는가? (높을수록! 🍌)
- Throughput(처리율) : 주어진 시간에 몇 개의 작업이 끝났는가? (높을수록! 🍌)
- Turnaround time(반환시간) : 작업을 시작한 순간부터 다 끝내고 나오는 시간 --> 어떤 작업이 Ready Queue에서 기다리다가 CPU에서 실행 후 다시 Ready Queue로 돌아가고 다시 반복... 하다가 어느 순간 완전히 작업이 끝났을 때! (짧을수록! 🍌)
- Waiting time(대기시간) : CPU서비스를 받기 위해서 Ready Queue에서 얼마나 기다렸는가? --> 단위는 sec (짧을수록! 🍌)
- Response time(응답시간) : 명령을 내리면 처음 응답이 나올 때까지 걸리는 시간(ex) 사이트 로딩 시간 (짧을수록! 🍌)

--> 병원으로 예를 들자면..

- CPU 이용률 : 병원장 입장에서 의사가 돈을 많이 받으니 받은 만큼 진료를 많이 하기를 원한다.
- 처리율 : 하루에 몇 명의 환자를 받는가?
- 반환시간 : 환자가 병원에 발을 들이는 순간부터 진료를 끝내고 나오는 시간
- 대기시간 : 의사를 만나기까지 환자가 대기하는 시간

CPU Scheduling Algorithms

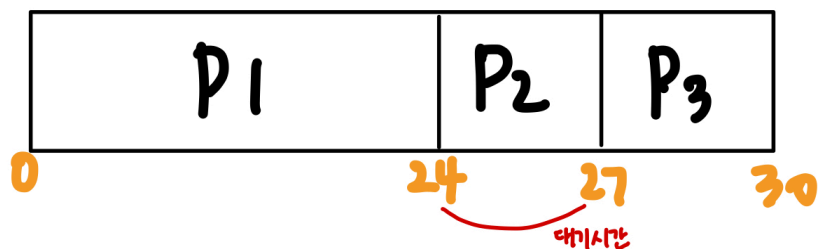
- First-Come, First-Served(FCFS) : 먼저 온 것을 먼저 서비스한다.
- Shortest-Job-First(SJF) : 작업시간이 짧은 것을 먼저 서비스한다.
- Priority : 우선순위가 높은 것을 먼저 서비스한다.
- Round-Robin(RR) : 빙빙 돌면서 순서대로 서비스한다.
- Multilevel Queue : Queue를 여러 개 둔다.
- Multilevel Feedback Queue : Queue를 여러 개 둔다.

First-Come, First-Served

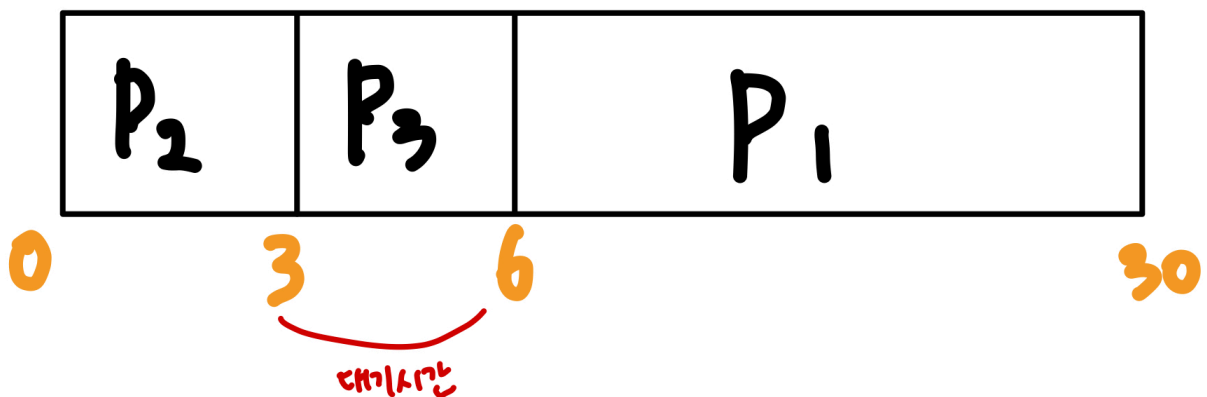
- 병원 & 은행 등 보통 이러한 방식이다.
- Simple & Fair : 가장 간단하고 일반적으로 가장 공정한 방법 --> 그러나 이 방법이 가장 좋다고는 할 수 없다.
- 예) Find Average Waiting Time - 대기 시간의 평균을 통해 효율적인지를 확인해볼 수 있다.
 - $AWT = (0+24+27)/3 = 17 \text{ msec}$
- Gantt Chart : 아래 그림과 같이 표현한 것(Gantt라는 사람의 이름을 따왔다)

Process	Burst Time (msec) - 사용하는 시간
P1	24
P2	3
P3	3

--> 예를 들자면, 은행에 P1, P2, P3 동시에 도착했는데 번호표를 P1-P2-P3 순으로 뽑게되었다.



--> 만약 순서가 P2-P3-P1이라면?



대기시간은 0msec, 3msec, 6msec가 된다. AWT를 계산하면, $(0+3+6) / 3 = 3\text{msec}$ 가 된다.

- Convoy Effect (호위효과) : 뒤에 붙어서 호위하는 것 --> P1 같이 시간이 오래 걸리는 것이 앞에 있으면, 시간이 짧게 걸리는 것들이 뒤에서 계속 기다리면서 마치 호위하듯이 따라다니는 것

- Nonpreemptive scheduling : 비선점 스케줄링이다.

! 들어온 순서대로 진행하면 대기 시간 면에서 별로 좋지 않다는 것을 알 수 있다.