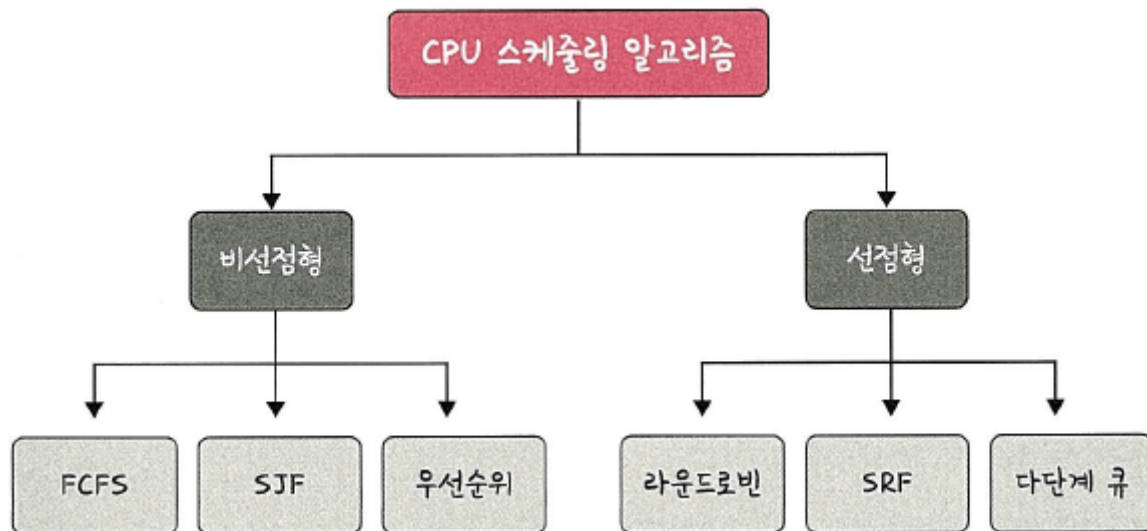


CPU 스케줄링 알고리즘



CPU 스케줄러는 CPU 스케줄링 알고리즘에 따라 프로세스에서 해야하는 일을 스레드 단위로 CPU에 할당

프로그램 실행 시에는 CPU 스케줄링 알고리즘이 어떤 프로그램에 CPU 소유권을 줄 것인지 결정함

알고리즘의 목표

- CPU 이용률을 높게함
- 주어진 시간에 많은 일을 하도록함
- 준비 큐 (Ready Queue)에 있는 프로세스는 적게함
- 응답시간은 짧게 설정함

비선점형 방식 (Non-Preemptive)

프로세스가 스스로 CPU 소유권을 포기하는 방식

강제로 프로세스를 중지하지 않으므로, 컨텍스트 스위칭으로 인한 부하가 적음

FCFS (First Come, First Served)

가장 먼저 온 것을 가장 먼저 처리하는 알고리즘

길게 수행되는 프로세스로 인해 준비 큐에서 오래 기다리는 현상 (Convoy Effect)이 발생하는 단점

SJF (Shortes Job First)

실행 시간이 가장 짧은 프로세스를 가장 먼저 실행

긴 시간을 가진 프로세스가 실행되지 않는 현상 (Starvation)이 일어남

평균 대기 시간이 가장 짧으나, 실제로는 실행 시간을 알 수 없으므로 과거의 실행시간을 토대로 추측해서 사용함

우선순위

기존 SJF 스케줄링의 경우, 긴 시간을 가진 프로세스가 실행되지 않는 현상이 발생

오래된 작업일수록 우선순위를 높이는 방법 (Aging)을 통해 SJF의 단점을 보완함

선점형 방식 (Preemptive)

현대 OS 가 채택한 방식

현재 사용 중인 프로세스를 알고리즘에 의해 중단시켜 버리고 강제로 다른 프로세스에 CPU 소유권을 할당하는 방식

라운드 로빈 (RR, Round Robin)

현대 컴퓨터가 쓰는 스케줄링인 우선순위 스케줄링의 일종

각 프로세스는 동일한 할당 시간을 주고 그 시간 안에 끝나지 않으면 다시 준비 큐의 뒤로 보냄

ex) q 만큼의 할당 시간이 부여되었고, N 개의 프로세스가 운영되는 경우

$(N-1) * q$ 의 시간이 지나면 자기 차례가 오게됨

할당 시간이 너무크면 FCFS가 되고, 짧으면 컨텍스트 스위칭이 잦아져 오버헤드, 즉 비용이 커짐

일반적으로 전체 작업 시간은 길어지나, 평균 응답 시간은 짧아진다는 특징이 있음

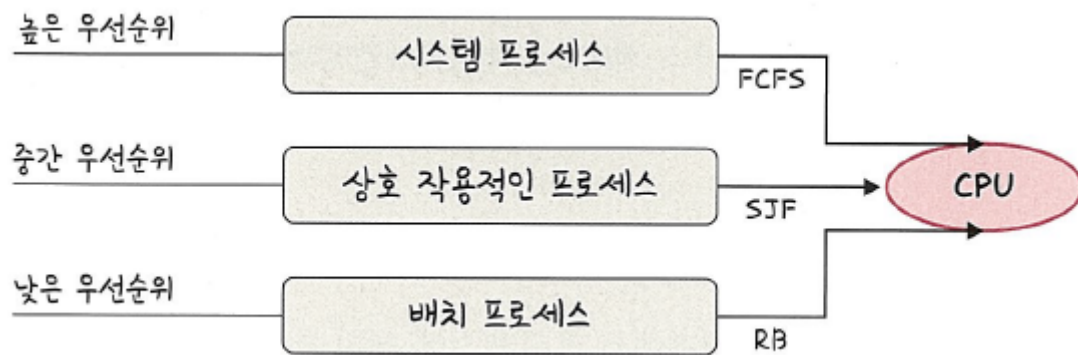
로드밸런서에서 트래픽 분산 알고리즘으로도 사용됨

SRF

중간에 더 짧은 작업이 들어오면, 수행하던 프로세스를 중지하고 해당 프로세스를 수행

SJF 는 중간에 실행 시간이 더 짧은 작업이 들어와도 기존의 작업을 모두 수행한 다음 해당 작업을 수행함

다단계 큐



우선순위에 따른 준비 큐를 여러개 사용

큐마다 라운드 로빈이나 FCFS 등 다른 스케줄링 알고리즘을 적용한 것

큐 간의 프로세스 이동이 안되므로, 스케줄링 부담이 적으나, 유연성이 떨어지는 특징이 있음