

과제_0419

스케줄링 알고리즘의 선택 기준

- 비선점형 알고리즘 : FCFS 스케줄링, SJF 스케줄링, HRN 스케줄링
- 선점형 알고리즘 : 라운드 로빈 스케줄링, SRT 스케줄링, 다단계 큐 스케줄링, 다단계 피드백 큐 스케줄링

선점형 알고리즘 -

실행 중인 프로세스를 강제로 중단시켜 다른 프로세스에게 CPU를 할당할 수 있는 알고리즘
이러한 방법으로 인해 더 높은 우선순위를 가진 프로세스가 CPU를 더 많이 사용할 수 있음
선점형 알고리즘은 시분할 시스템에서 사용됨

비선점형 알고리즘 -

실행 중인 프로세스가 종료되거나 대기 상태로 들어가기 전까지 다른 프로세스로의 전환이 발생하지 않는 알고리즘

이 방법은 배치 처리 시스템에서 사용됨

스케줄링 알고리즘의 평가 기준

- 대기 시간 : 프로세스가 생성된 후 실행되기 전까지 대기하는 시간
- 응답 시간 : 첫 작업을 시작한 후 첫 번째 출력이 나오기까지의 시간
- 실행 시간 : 프로세스 작업이 시작된 후 종료되기까지의 시간
- 반환 시간 : 대기 시간을 포함하여 실행이 종료될 때까지의 시간

SJF (Shortest Job First)

- 정의
 - 실행 시간이 가장 짧은 작업부터 처리하는 스케줄링 알고리즘
- 장점

- 평균 대기 시간이 적고, 작업을 빨리 처리할 수 있음
- 단점
 - 작업의 실행 시간을 정확하게 예측하기 어려워 예측 오차가 발생할 수 있음. 따라서, 긴 작업이 계속해서 들어올 경우, 무한정 기다리게 되는 문제가 발생할 수 있음
- 최적의 성능을 보이거나, 긴 작업이 대기하는 경우에는 평균 대기 시간이 길어질 수 있음
- 선점형(SJF Preemptive)과 비선점형(SJF Non-Preemptive)으로 나뉘어짐

SRT (Shortest Remaining Time)

- 정의
 - 현재 실행 중인 작업의 남은 실행 시간보다 짧은 실행 시간을 가진 작업이 들어올 경우, 그 작업을 먼저 처리하는 스케줄링 알고리즘
- 장점
 - SJF보다 평균 대기 시간이 더 적음. 실행 시간이 긴 작업이 먼저 들어올 경우, 작업의 남은 실행 시간을 감안하여 다른 작업을 처리하므로 무한정 기다리는 문제가 발생하지 않음
- 단점
 - 작업의 실행 시간을 정확하게 예측하기 어려워 예측 오차가 발생할 수 있음. 또한, 실행 시간이 짧은 작업이 지나치게 우선적으로 처리되는 경우, 실행 시간이 긴 작업의 대기 시간이 증가할 수 있음

RR (Round Robin)

- 정의
 - 시간 할당량(TQ, Time Quantum)을 정해두고, 할당된 시간 안에 작업을 처리하지 못한 경우, 다른 작업으로 전환하여 처리하는 스케줄링 알고리즘
- 장점
 - 모든 작업에게 공평한 처리 기회를 부여할 수 있음. 또한, 대화식 시스템에서는 사용자 응답 시간을 보장할 수 있음, 응답 시간이 빠르고 선점형 알고리즘이므로 긴 작업이 대기하는 경우에도 다른 작업들이 처리될 수 있음
- 단점
 - 시간 할당량을 정하는 것이 중요하며, 할당량이 작으면 문맥 교환(context switching) 오버헤드가 증가하여 성능이 저하될 수 있음. 또한, 실행 시간이 긴 작업이 먼저 들어올 경우, 평균 대기 시간이 길어질 수 있음

HRRN (Highest Response Ratio Next)

- 정의
 - 대기 시간과 작업의 실행 시간을 고려하여 우선순위를 계산하고, 우선순위가 높은 작업을 먼저 처리하는 스케줄링 알고리즘
- 장점
 - 대기 시간과 실행 시간을 모두 고려하기 때문에, SJF나 SRT와 같은 알고리즘보다 평균 대기 시간이 짧을 수 있음. 또한, 긴 작업이 먼저 들어와도 대기 시간을 줄일 수 있음
- 단점
 - 우선순위 계산에 따라 오버헤드가 증가할 수 있음. 또한, 작업의 대기 시간이 너무 길어지는 경우, 우선순위가 높아져 처리될 가능성이 높아지므로, 긴 작업이 계속해서 들어올 경우, 무한정 기다리는 문제가 발생할 수 있음

MFQ (Multi-Level Feedback Queue)

- 정의
 - 여러 개의 우선순위 큐를 사용하여 작업을 처리하는 스케줄링 알고리즘. 각 큐는 서로 다른 우선순위와 시간 할당량을 가지며, 작업이 큐에서 이동하면 우선순위와 시간 할당량이 변경될 수 있음
- 장점
 - 우선순위와 시간 할당량을 조절할 수 있어, 다양한 작업의 처리를 유연하게 할 수 있음. 또한, 작업의 종류와 특성에 따라 다른 큐에서 처리할 수 있어, 성능을 최적화할 수 있음
- 단점
 - 큐의 우선순위와 시간 할당량을 설정하는 것이 중요하며, 적절한 설정이 이루어지지 않으면 평균 대기 시간이 길어질 수 있음, 큐의 수가 많아지면 오버헤드가 증가할 수 있음

에이징 (나이 먹기)

- 아사 현상의 완화 방법
- 프로세스가 양보할 수 있는 상한선을 정하는 방식

- 프로세스가 자신의 순서를 양보할 때마다 나이를 한 살씩 먹어 최대 몇 살까지 양보하도록 규정하는 것

라운드 로빈 스케줄링의 동작 방식

- 한 프로세스가 할당받은 시간 (타임 슬라이스) 동안 작업을 하다가 작업을 완료하지 못하면 준비 큐의 맨 뒤로 가서 자기 차례를 기다리는 방식
- 선점형 알고리즘 중 가장 단순하고 대표적인 방식
- 프로세스들의 작업을 완료할 때까지 계속 순환하면서 실행
- 타임 슬라이스의 크기와 문맥 교환
 - 라운드 로빈 스케줄링이 효과적으로 작동하려면 문맥 교환에 따른 추가 시간을 고려하여 타임 슬라이스를 적절히 설정해야 함
- 타임 슬라이스가 큰 경우
 - 하나의 작업이 끝난 뒤 다음 작업이 시작되는 것처럼 보여 FCFS 스케줄링과 다를 게 없음
- 타임 슬라이스가 작은 경우
 - 문맥 교환이 너무 자주 일어나 문맥 교환에 걸리는 시간이 실제 작업 시간보다 상대적으로 커지며, 문맥 교환에 많은 시간을 낭비하여 실제 작업을 못하는 문제가 발생
- 타임 슬라이스는 되도록 작게 설정하되 문맥 교환에 걸리는 시간을 고려하여 적당한 크기로 하는 것이 중요함
- 유닉스 운영체제에서는 타임 슬라이스가 대략 100 밀리초

다단계 큐 스케줄링

- 동작 방식
 - 우선순위에 따라 준비 큐를 여러 개 사용하는 방식
 - 프로세스는 운영체제로부터 부여받은 우선순위를 따라 해당 우선순위의 큐에 삽입
 - 우선순위는 고정형 우선순위 사용

