

나올 수 있는 면접 질문

cpu 스케줄링이란 무엇인가요?

CPU 스케줄링은 프로세스가 작업을 수행할 때, 언제 어떤 프로세스에 cpu를 할당할지 결정하는 작업이다.

cpu 스케줄링의 목적은 무엇인가요?

공평성, 효율성, 안정성, 확장성, 반응시간보장, 무한연기방지 이다.

스케줄링시에 고려해야할 사항은 무엇이 있을까요?

1. 선점형과 비선점형
2. 프로세스의 우선순위
3. CPU 집중 프로세스와 입출력 집중 프로세스
4. 전면 프로세스와 후면 프로세스

CPU 스케줄링

CPU 스케줄링은 프로세스가 작업을 수행할 때, 언제 어떤 프로세스에 cpu를 할당할지 결정하는 작업이다. 식당의 식당 관리자가 좌석을 관리하거나, 조리 순서를 관리하는 것 처럼, CPU도 프로세스들을 어떤 것 부터 실행할지, 프로세스 몇개를 작업할지를 결정한다.

CPU 스케줄링의 단계 3가지



CPU 스케줄링의 단계는 크게 고수준, 저수준, 중간 수준의 스케줄링으로 나뉜다.

고수준 스케줄링

고수준 스케줄링은 가장 큰 틀에서 이루어지는 스케줄링이며, 장기 스케줄링, 작업 스케줄링, 승인 스케줄링이라고도 한다.

고수준 스케줄링은 전체 시스템의 부하를 고려하여 작업을 시작할지 말지를 결정한다. 이 결정에 따라 시스템의 전체 프로세스 수가 결정되는데 이를 멀티프로그래밍 정도라고 한다.

중간수준 스케줄링

시스템의 부하를 조절하기 위해서는 중간 수준 스케줄링을 고려한다. 시스템에 과부하가 걸려 전체 프로세스 수를 조절해야 한다면, 이미 활성화된 프로세스중 일부를 보류 상태로 보낸다. 보류된 프로세스는 처리 능력에 여유가 생기면 다시 활성화한다.

저수준 스케줄링

저수준 스케줄링에서는 실제로 작업이 이루어진다. 오늘날의 cpu 스케줄러가 대부분 중간 수준 스케줄링과 저수준 스케줄링으로 구성되어 있다. 필요에 따라 준비 상태에 있는 프로세스를 실행 상태로 옮기기도 하고, 대기 상태로 보내기도 하며, 타임 아웃으로 준비 상태로 돌려보내기도 한다.

CPU 스케줄링의 목적 6가지

공평성

모든 프로세스가 자원을 공평하게 배정받아야 하며, 자원 배정 과정에서 특정 프로세스가 배제되어서는 안된다.

효율성

시스템 자원이 유휴 시간 없이 사용되도록 스케줄링을 하고, 유휴 자원을 사용하려는 프로세스에는 우선권을 주어야 한다.

안정성

우선순위를 사용하여 중요 프로세스가 먼저 작동하도록 배정함으로써 시스템 자원을 점유하거나 파괴하려는 프로세스로부터 자원을 보호해야한다.

확장성

프로세스가 증가해도 시스템이 안정적으로 작동하도록 조치해야 한다. 또한 시스템 자원이 늘어나는 경우 이 혜택이 시스템에 반영되게 해야 한다.

반응 시간 보장

응답이 없는 경우 사용자는 시스템이 멈춘 것으로 가정하기 때문에 시스템은 적절한 시간 안에 프로세스의 요구에 반응해야 한다.

무한 연기 방지

특정 프로세스의 작업이 무한히 연기되어서는 안된다

CPU 스케줄링 평가 기준 5가지

CPU 이용률(CPU utilization)

시간당 CPU를 사용한 시간의 비율

프로세서를 실행상태로 항상 유지하려고 해야 한다.

처리율(Throughput)

시간당 처리한 작업의 비율

단위 시간당 완료되는 작업 수가 많도록 해야 한다.

반환시간(Turnaround Time)

프로세스가 생성된 후 종료되어 사용하던 자원을 모두 반환하는 데까지 걸리는 시간

작업이 준비 큐(ready queue)에서 기다린 시간부터 CPU에서 실행된 시간, I/O 작업 시간의 합이다.

대기시간(Waiting Time)

대기열에 들어와 CPU를 할당받기 까지 기다린 시간

준비 큐에서 기다린 시간의 총합

반응시간(Response Time)

대기열에서 처음으로 CPU를 얻을 때까지 걸린 시간

대기시간과 비슷하지만 다른 점은, 대기시간은 준비 큐에서 기다린 모든 시간을 합친 것이지만 반응 시간은 CPU를 할당받은 최초의 순간까지 기다린 시간 한번 만을 측정한다.

> -> CPU 이용률과 처리율은 극대화하는 것이 좋고 반환시간, 대기시간, 반응시간은 줄이는 것이 일반적으로 좋다.

스케줄링시 고려 사항 4가지

CPU 스케줄러가 어떤 프로세스에 우선적으로 CPU를 할당할지 결정할 때 고려할 사항이 몇가지 있다.

선점형 스케줄링과 비선점형 스케줄링

선점형 스케줄링 : 어떤 프로세스가 CPU를 할당받아 실행 중이라도 운영체제가 CPU를 강제로 빼앗을 수 있는 스케줄링 방식이다.

비선점형 스케줄링 : 어떤 프로세스가 CPU를 점유하면 다른 프로세스가 이를 빼앗을 수 없는 스케줄링 방식이다.

구분	선점형	비선점형
작업방식	실행 상태에 있는 작업을 중단시키고 새로운 작업을 실행할 수 있다	실행 상태에 있는 작업이 완료될 때까지 다른 작업이 불가능하다
장점	프로세스가 cpu를 독점할 수 없어 대화형이나 시분할 시스템에 적합하다.	cpu스케줄러의 작업량이 적고 문맥 교환의 오버헤드가 적다.
단점	문맥 교환의 오버헤드가 많다.	기다리는 프로세스가 많아 처리율이 떨어진다.
사용	시분할 방식 스케줄러에 사용된다.	일괄 작업 방식 스케줄러에 사용된다.
중요도	높다	낮다

프로세스 우선순위

대부분의 cpu 스케줄러는 우선순위를 사용한다. 우선순위가 있다는 것은 프로세스의 중요도가 다르다는 의미이다. 프로세스는 크게 커널 프로세스와 일반 프로세스로 나뉘는데 커널 프로세스의 우선순위가 일반 프로세스보다 더 높다. 커널에 관련된 작업이 더 중요하다는 뜻이기도 하다.

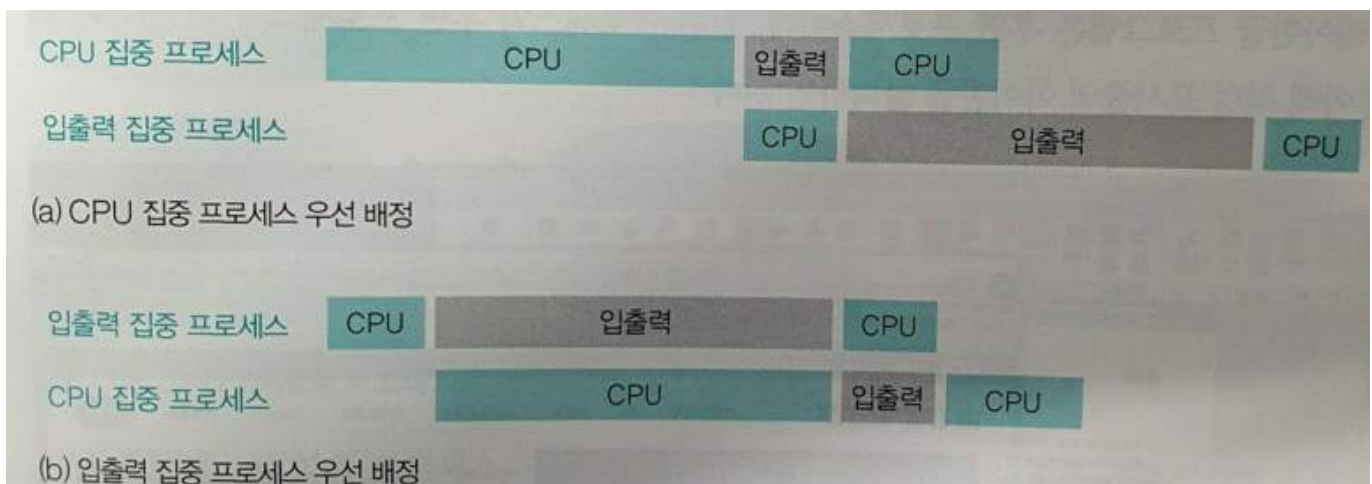
cpu 집중 프로세스와 입출력 집중 프로세스

cpu 집중 프로세스 : 수학 연산과 같이 cpu를 많이 사용하는 프로세스를 말한다. 즉 cpu버스트가 많은 프로세스이다.

> cpu 버스트 : 프로세스는 생성된 후 준비, 실행, 대기 상태를 거쳐 완료되는데 프로세스가 cpu를 할당받아 실행하는 작업을 cpu 버스트라고 한다.

입출력 집중 프로세스 : 저장장치에서 데이터를 복사하는 일과 같이 입출력을 많이 사용하는 프로세스를 말한다. 즉 입출력 버스트가 많은 프로세스이다.

> 입출력 버스트 : 프로세스가 입출력 작업을 하는 것을 입출력 버스트라고 한다.



두개의 집중 프로세스가 같이 있을 때는 입출력 집중 프로세스를 먼저 실행상태로 옮기는 것이 효율적이다. 입출력 집중 프로세스는 잠깐 cpu를 사용한 이후 대기 상태로 이동하기 때문에 다른 프로세스가 cpu를 사용

할 수 있기 때문이다.

전면 프로세스와 후면 프로세스

전면 프로세스 : GUI를 사용하는 운영체제에서 화면의 맨 앞에 놓인 프로세스를 말한다. 현재 입력과 출력을 사용하는 프로세스이며, 사용자와 상호작용이 가능하여 상호작용 프로세스라고도 한다.

후면 프로세스 : 사용자와 상호작용이 없는 프로세스이다. 압축 프로그램처럼 사용자의 입력 없이 작동하기 때문에 일괄 작업 프로세스라고도 한다.

전면 프로세스는 사용자의 요구에 즉각 반응해야 하지만 후면 프로세스는 상호작용이 없다. 따라서 전면 프로세스의 우선순위가 후면 프로세스보다 높다. 그만큼 후면 프로세스는 전면 프로세스보다 CPU를 할당 받을 확률이 적다.