

CPU 스케줄링, 컴퓨터의 똑똑한 교통 경찰 이야기

1. 컴퓨터 세상의 교통 경찰, CPU 스케줄링이란?

컴퓨터 내부를 하나의 '바쁜 도시'라고 상상해 봅시다. 이 도시에는 수많은 '자동차(프로세스)'들이 저마다의 목적지를 향해 동시에 달려가고 싶어 합니다. 하지만 이 도시에는 모든 자동차가 반드시 지나가야만 하는 '하나뿐인 초고속 도로(CPU)'가 있습니다. 만약 아무런 규칙 없이 모든 자동차가 이 도로에 한꺼번에 진입하려고 한다면 어떻게 될까요? 도시는 순식간에 극심한 정체와 혼란에 빠질 것입니다. 바로 이 지점에서 '교통 정리'의 필요성이 대두됩니다. 이러한 혼란을 막기 위해 도시에는 '교통 경찰(운영체제)'이 존재합니다. 이 교통 경찰이 하는 일이 바로 **CPU** 스케줄링입니다. **CPU** 스케줄링이란 운영체제가 프로세스들에게 공정하고 합리적으로 **CPU**라는 자원을 배분하는 규칙 또는 정책을 의미합니다. 만약 이 스케줄링이 제대로 이루어지지 않으면, 컴퓨터는 아주 무질서한 상태에 빠지게 됩니다. 예를 들어, 지금 당장 실행되어야 할 중요한 프로세스는 계속 기다리기만 하고, 급하지 않은 프로세스들만 **CPU**를 독점하는 상황이 발생할 수 있습니다. 이는 컴퓨터 전체의 성능과 직결되는 매우 중요한 문제입니다. 이제 이 똑똑한 교통 경찰이 왜 필요하고, 또 어떤 기준으로 교통 정리를 하는지 더 자세히 알아보겠습니다.

2. 모두에게 공평한 것이 최선일까? '우선순위'의 등장

가장 단순하고 공정해 보이는 방법은 무엇일까요? 아마 "CPU를 사용하고 싶다고 먼저 손든 프로세스 순서대로 차례대로 이용하게 하는" 방식일 것입니다. 하지만 언뜻 합리적으로 보이는 이 방법은 실제로는 그리 좋은 방법이 아닙니다. 왜냐하면 도시를 달리는 자동차들이 모두 같지 않듯, 컴퓨터의 프로세스들도 모두 동등하지 않기 때문입니다. 어떤 프로세스는 구급차처럼 빨리 처리해야 하고, 어떤 프로세스는 장거리 화물 트럭처럼 천천히 처리해도 괜찮습니다. 바로 이 개념이 **CPU** 스케줄링의 핵심인 ***'우선순위(Priority)'"입니다. 우선순위는 더 효율적인 자원 배분을 위한 핵심 기준으로, 운영체제는 이 우선순위를 바탕으로 어떤 프로세스에 **CPU**를 먼저 할당할지 결정합니다.

핵심 예시: 구급차와 화물 트럭의 차이 (**I/O** 집중 vs. **CPU** 집중)

프로세스의 우선순위는 '입출력(I/O) 집중 프로세스'와 'CPU 집중 프로세스'의 구분을 통해 가장 잘 이해할 수 있습니다. 이를 구급차와 화물 트럭에 비유해 봅시다. | 구분 | **I/O** 집중 프로세스 (구급차) | **CPU** 집중 프로세스 (화물 트럭) | ----- | ----- | ----- | 특징 | 구급차: 초고속 도로(CPU)는 병원(입출력 장치)으로 가기 위해 잠깐만 필요함. 대부분의 시간을 병원에서 보냄. | 화물 트럭: 목적지(예산 완료)까지 초고속 도로(CPU) 위에서 계속 달려야 함. | 예시 | 비디오 재생, 디스크 백업 | 복잡한 수학 연산, 컴파일, 그래픽 계산 | 우선순위 | 높음 | 상대적으로 낮음 | 이유 | 구급차를 먼저 보내주면 초고속 도로가 금방 비게 되어, 다른 차량(특히 화물 트럭)이 도로를 더 오래, 효율적으로 사용할 수 있음. | 화물 트럭은 도로를 오래 차지하지만, 구급차만큼 긴급하지 않으므로 잠시 기다려도 전체 교통 흐름에 더 이득임. |

이처럼 운영체제는 각 프로세스의 특성을 파악하여 더 효율적인 방향으로 우선순위를 부여합니다. 이러한 우선순위 정보는 각 프로세스의 신분증과도 같은 **PCB(프로세스 제어 블록)**에 저장되어 스케줄링에 활용됩니다. 그렇다면 운영체제는 이 우선순위를 바탕으로 수많은 프로세스들을 어떻게 효율적으로 관리하고 줄을 세우는지, 그 비밀을 파헤쳐 보겠습니다.

3. 프로세스들의 대기실, 스케줄링 큐

수많은 프로세스가 CPU를 사용하겠다고 아우성치는 상황에서, 운영체제가 다음 차례를 정하기 위해 일일이 모든 프로세스의 PCB를 뒤져보는 것은 매우 비효율적입니다. 그래서 운영체제는 **스케줄링 큐(Scheduling Queue)**라는 아주 스마트한 방법을 사용합니다. 스케줄링 큐란 특정 자원(CPU, 하드디스크 등)을 사용하고 싶어 하는 프로세스들이 서는 '가상의 줄'입니다. 마치 놀이공원에서 인기 있는 놀이기구를 타기 위해 줄을 서는 것처럼, 프로세스들도 자신이 사용하고 싶은 자원별로 마련된 큐에 서서 자기 차례를 기다립니다. 한 가지 중요한 점은, 스케줄링 큐가 자료구조 '큐'와 이름이 같다고 해서 반드시 먼저 들어온 순서(**FIFO**)대로 처리되는 것은 아니라는 점입니다. 앞서 배운 '우선순위'가 더 결정적인 요소로 작용하여, 줄의 중간에 있더라도 우선순위가 높은 프로세스가 먼저 CPU를 할당받을 수 있습니다.

대표적인 스케줄링 큐

가장 대표적인 스케줄링 큐는 다음과 같습니다.

- **준비 큐 (Ready Queue):** CPU를 사용하기 위해 '준비 상태'에 들어선 프로세스들이 대기하는 줄입니다.
- **대기 큐 (Waiting Queue):** 입출력 작업을 요청하고 '대기 상태'에 들어선 프로세스들이 대기하는 줄입니다. 특히 대기 큐는 하나만 있는 것이 아니라, 하드디스크 전용 대기 큐, 프린터 전용 대기 큐처럼 각 입출력 장치별로 여러 개가 존재하는 경우가 많습니다. 덕분에 같은 장치를 기다리는 프로세스들끼리 효율적으로 줄을 설 수 있습니다.

프로세스의 여정: 큐 사이의 이동

프로세스는 컴퓨터가 꺼질 때까지 다음과 같은 흐름에 따라 큐 사이를 끊임없이 이동합니다.

1. **준비 큐로 이동:** 실행 준비가 된 프로세스들은 '준비 큐'에서 자기 차례를 기다립니다 (준비 상태).
2. **CPU 할당 (실행):** 자기 차례가 되면 CPU를 할당받아 실행됩니다 (실행 상태).
3. 다시 준비 큐로: 할당된 시간이 끝나면(타이머 인터럽트), 다시 '준비 큐'의 끝으로 돌아가 다음 차례를 기다립니다.
4. **대기 큐로 이동:** 실행 중 입출력 작업이 필요해지면, CPU를 반납하고 해당 입출력 장치의 '대기 큐'로 이동합니다 (대기 상태).
5. **입출력 완료 후 준비 큐로:** 입출력 작업이 완료되면(입출력 완료 인터럽트), '대기 큐'에서 나와 다시 '준비 큐'로 이동하여 CPU 할당을 기다립니다. 이처럼 프로세스들을 관리하는 방식에는 크게 두 가지 스타일이 있습니다. 한 번 CPU를 잡으면 끝까지 쓰는 방식과, 필요에 따라 중간에 빼앗을 수 있는 방식에 대해 알아봅시다.

4. 두 가지 스케줄링 스타일: 선점형 vs. 비선점형

"어떤 화물 트럭이 초고속 도로를 달리고 있을 때, 뒤에서 사이렌을 울리는 구급차가 나타나면 어떻게 해야 할까요?" 이 질문에 대한 답이 바로 두 가지 스케줄링 스타일의 핵심 차이를 보여줍니다. | 구분 | 선점형 스케줄링 (**Preemptive**) | 비선점형 스케줄링 (**Non-preemptive**) || ----- | ----- | ----- || 핵심 개념 | 교통 경찰(운영체제)이 화물 트럭을 강제로 간길에 세우고 구급차를 먼저 보낼 수 있음. 즉, 실행 중인 프로세스로부터 CPU를 빼앗아 더 높은 우선순위의 프로세스에 할당할 수 있습니다. | 화물 트럭이 한번 도로에 진입하면, 스스로 목적지(작업 종료)나 휴게소(대기 상태)에 도착하기 전까지는 절대 멈추게 할 수 없음. 구급차는 뒤에서 계속 기다려야 합니다. || 장점 | • 어느 한 프로세스의

자원 독점을 막고, 자원을 골고루 배분할 수 있음. | • 문맥 교환이 적게 발생하여 오버헤드가 적음. || 단점 | • 문맥 교환이 자주 발생하여 오버헤드가 클 수 있음. | • 모든 프로세스가 골고루 자원을 이용하기 어려움. (하나의 긴 작업이 모두를 기다리게 할 수 있음)|
우리가 일반적으로 사용하는 현대의 운영체제는 대부분 선점형 스케줄링 방식을 채택합니다. 프로세스들에게 정해진 시간만큼 CPU를 사용하게 하고, 시간이 다 되면 CPU를 회수하여 다른 프로세스에 넘겨주는 방식이 바로 선점형 스케줄링의 대표적인 예입니다.

5. 정리: 똑똑한 교통 경찰 덕분에 쾌적한 컴퓨터 환경

지금까지 컴퓨터의 성능을 좌우하는 핵심 원리인 CPU 스케줄링에 대해 알아보았습니다.
핵심 개념을 다시 한번 정리해 보겠습니다.

- **CPU 스케줄링:** 한정된 CPU 자원을 여러 프로세스에 효율적으로 배분하는 운영체제의 역할.
- **우선순위:** 더 중요하거나 빨리 처리해야 할 작업을 먼저 실행하기 위한 기준.
- **스케줄링 큐:** 프로세스들이 자원을 기다리며 대기하는 줄 (준비 큐, 대기 큐).
- **선점형/비선점형:** CPU를 강제로 빼앗을 수 있는지 여부에 따른 두 가지 스케줄링 스타일. 결론적으로 CPU 스케줄링은, **'우선순위'**라는 교통 법규에 따라 **'큐'**라는 지정된 차선에서 대기 중인 프로세스(자동차)들을 '선점형' 또는 '비선점형' 스타일로 지휘하여, CPU라는 단 하나의 초고속 도로를 모두가 효율적으로 사용하게 만드는 똑똑한 교통 경찰과 같습니다. 이 글에서 배운 기본 개념들은 앞으로 여러분이 마주할 FCFS, SJF, 라운드 로빈 등 다양한 'CPU 스케줄링 알고리즘'을 이해하는 데 매우 중요한 기초 지식이 될 것입니다.