

● Deadlock에 대해서 간단히 설명해라. / Deadlock은 언제 발생하게 되나?

▶ 교착 상태(Deadlock)란 둘 이상의 thread가 각기 다른 thread가 점유하고 있는 자원을 서로 기다릴 때, 무한 대기 에 빠지는 상황을 말합니다. deadlock은 상호 배제(mutual exclusion), 점유 대기(hold-and-wait), 비선점(no preemption), 순환 대기(circular wait)의 4가지 조건이 동시에 성립할 때 발생할 수 있습니다. 상호 배제는 동시에 한 thread만 자원을 점유할 수 있는 상황이고 (다른 thread가 자원을 사용하려면 자원이 방출될 때까지 기다려야 합니다), 점유 대기는 thread가 자원을 보유한 상태에서 다른 thread가 보유한 자원을 추가적으로 기다리는 상황입니다. 또 비선점은 다른 thread가 사용 중인 자원을 강제로 선점할 수 없는 상황을 뜻하고 (자원을 점유하고 있는 thread에 의해서만 자원이 방출됩니다), 순환 대기는 대기 중인 thread들이 순환 형태로 자원을 대기하는 상황을 말합니다. deadlock 문제를 해결하는 방법에는 무시, 예방, 회피, 탐지-회복의 4가지 방법이 있습니다

기법	설명	비고
무시	deadlock 발생 확률이 낮은 시스템에서 아무런 조치도 취하지 않고 deadlock을 무시하는 방법입니다.	무시 기법은 시스템 성능 저하가 없다 는 큰 장점이 있습니다. 현대 시스템에서는 deadlock이 잘 발생하지 않고, 해결 비용이 크기 때문에 무시 방법이 많이 사용됩니다.
예방	교착 상태의 4가지 발생 조건중 하나가 성립하지 않게 하는 방법입니다.	순환 대기 조건이 성립하지 않도록 하는 것이 현실적으로 가능한 예방 기법입니다. 자원 사용의 효율성이 떨어지고 비용이 큼 니다.
회피	thread가 앞으로 자원을 어떻게 요청 할지에 대한 정보를 통해 순환 대기 상태가 발생하지 않도록 자원을 할당 하는 방법입니다.	자원 할당 그래프 알고리즘, 은행원 알고리즘 등을 사용하여 자원을 할당 하여 deadlock을 회피합니다.
탐지-회복	시스템 검사를 통해 deadlock 발생을 탐지하고, 이를 회복시키는 방법입니다.	자원 사용의 효율성이 떨어지고 비용이 큼 니다.