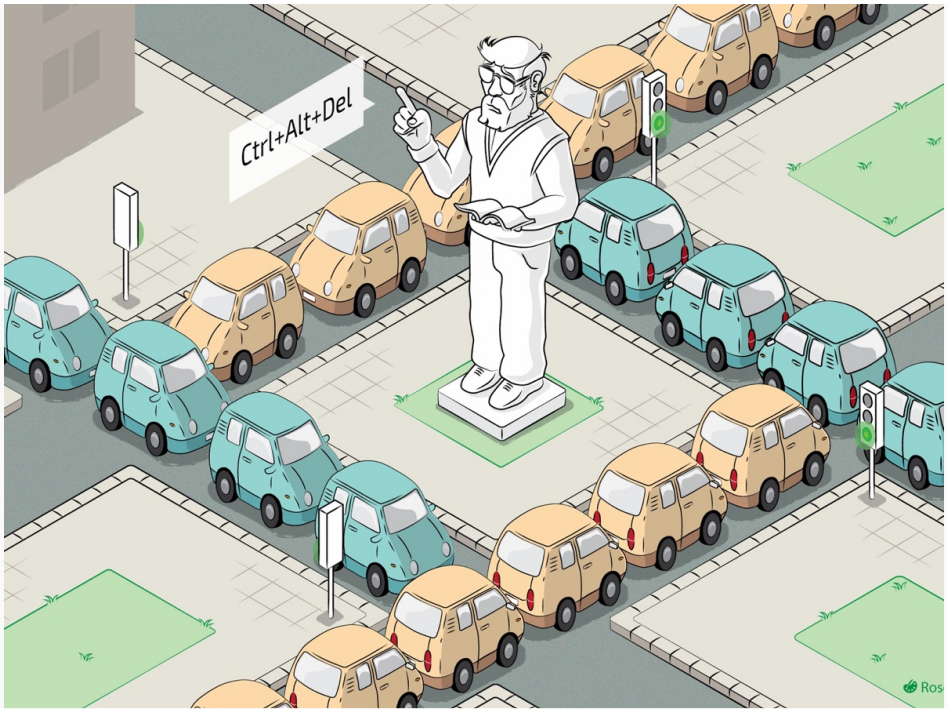
데드락(Deadlock, 교착 상태)



프로세스나 스레드가 서로 상대방의 자원을 점유하고 있는 상태에서 더 이상 진행할 수 없는 상태를 말합니다. 이러한 상태에서는 각각의 프로세스나 스레드가 다른 프로세스나 스레드가 점유한 자원을 기다리며 아무 일도 처리할 수 없게 됩니다.

데드락이 발생하면 시스템이 정지되어 모든 프로세스나 스레드가 종료되기 전까지 대기 상태로 남아 있게 됩니다. 이러한 상황은 시스템의 안정성과 가용성을 저하시키며, 대규모 시스템에서 데드락이 발생하면 심각한 문제를 초래할 수 있습니다.

발생 원인

상호 배제(Mutual Exclusion)

자원은 한 번에 한 프로세스나 스레드에 의해 점유될 수 있습니다. 이러한 상태에서 다른 프로세스나 스레드가 해당 자원을 점유하기 위해 대기하게 됩니다.

점유 대기(Hold and Wait)

하나의 프로세스나 스레드가 이미 자원을 점유한 상태에서 다른 자원을 요청하면서 해당 자원을 점유하기 위해 대기하는 상태를 말합니다.

비선점(Non-preemptive) 상태

다른 프로세스나 스레드가 점유한 자원을 강제로 빼앗을 수 없는 상태입니다. 이러한 상태에서 다른 프로세스나 스레드가 해당 자원을 사용하기 위해 대기하게 됩니다.

순환 대기(Circular Wait)

자원을 점유한 프로세스나 스레드들이 서로 자원을 기다리며 막힌 상태를 말합니다.

해결방안

예방(prevention)

데드락이 발생하지 않도록 하는 방법입니다. 이 방법은 데드락 발생의 조건인 상호 배제, 점유 대기, 비선점, 순환 대기 중 하나 이상을 제거하여 데드락이 발생하지 않도록 합니다. 하지만 이 방법은 리소스 사용의 효율성이 떨어지고, 모든 상황에서 데드락을 예방할 수 없다는 한계가 있습니다.

회피(avoidance)

데드락이 발생하지 않을 때까지 프로세스의 자원 요구에 대한 부가적인 정보(자원 요구량, 사용 중인 자원 등)를 이용하여 시스템이 데드락 발생 가능성을 사전에 예측하고 회피하는 방법입니다. 이 방법은 예방 방법에 비해 더욱 효율적으로 리소스를 사용할 수 있지만, 데드락 발생 가능성을 사전에 예측하고 회피하기 위한 추가적인 자원과 알고리즘이 필요합니다.

탐지 및 회복(detection and recovery)

데드락이 발생하면, 시스템이 이를 탐지하여 데드락을 해결하는 방법입니다. 이 방법은 예방 및 회피 방법에 비해 비용이 적게 들지만, 데드락이 이미 발생한 후에 해결이 이루어지므로 일시적인 시스템 성능 저하가 발생할 수 있습니다.

무시(ignorance)

데드락을 일으키는 조건이나 확률이 매우 낮은 경우에는 데드락 발생 가능성을 무시하는 방법입니다. 하지만 이 방법은 위험부담이 매우 크기 때문에 사용되지 않거나, 매우 드문 경우에만 사용됩니다.