

13장 교착 상태

교착상태

- 일어나지 않을 사건을 기다리며 진행이 멈춰 버리는 현상

교착상태 발생 조건

- 상호 배제
 - 한 프로세스가 사용하는 자원을 다른 프로세스가 사용할 수 없는 상태
- 점유와 대기
 - 자원을 할당받은 상태에서 다른 자원을 할당받기를 기다리는 상태
- 비선점
 - 자원을 이용하는 프로세스의 작업이 끝나야만 비로소 이용할 수 있는 것
- 원형 대기
 - 프로세스들이 원의 형태로 자원을 대기하는 것

교착 상태 해결 방법

교착 상태 예방

- 프로세스들에 자원을 할당할 때 상호 배제, 점유와 대기, 비선점, 원형 대기 중 하나의 조건이라도 만족시키지 않게 할당하면 교착 상태는 발생하지 않음
- 상호 배제를 없애는 방법
 - 뭉든 자원을 공유 가능하게 만든다는 말
 - 현실적으로 사용하기에는 다소 무리가 있음
- 점유와 대기를 없애는 방법
 - 프로세스에 자원을 모두 할당하거나, 아예 할당하지 않는 방식으로 배분
 - 단점
 - 자원의 활용률이 낮아짐 : 당장 자원이 필요해도 기다릴 수밖에 없는 프로세스와 사용되지 않으면서 오랫동안 할당되는 자원을 다수 양산하기 때문에 자원의 활용률이 낮아짐.

- 자원을 많이 사용하는 프로세스는 자원을 적게 사용하는 프로세스에 비해 동시에 자원을 사용할 타이밍을 확보하기가 어려워져서, 결국 자원을 많이 필요로 하는 프로세스가 무한정 기다리게 되는 기아 현상을 야기할 우려가 있음
- **비선점 조건을 없애는 방법**
 - 모든 자원을 빼앗을 수 있도록 하여 교착 상태를 예방하는 방법
 - 단점
 - 프린터의 경우 도중에 다른 프로세스가 프린터 자원을 빼앗아 사용하면 문제가 발생
 - 한 프로세스의 작업이 끝날 때까지 다른 프로세스가 기다려야하는 자원도 얼마든지 있음
- **원형 대기 조건을 없애는 방법**
 - 모든 자원에 번호를 붙이고, 오름차순으로 자원을 할당하면 원형 대기는 발생하지 않음
 - 단점
 - 모든 컴퓨터 시스템 내에 존재하는 수많은 자원에 번호를 붙이는 일은 그리 간단한 작업이 아님
 - 각 자원에 어떤 번호를 붙이는지에 따라 특정 자원의 활용률이 떨어질 수 있음

교착 상태 회피

- 프로세스들에 배분할 수 있는 자원의 양을 고려하여 교착 상태가 발생하지 않을 정도의 양만큼만 자원을 배분하는 방법이 교착 상태 회피
 - 안전 상태를 유지할 수 있는 경우에만 자원을 할당하는 방법
- **안전 상태** : 교착 상태가 발생하지 않고 모든 프로세스가 정상적으로 자원을 할당받고 종료될 수 있는 상태
- **불안정 상태** : 교착 상태가 발생할 수도 있는 상황
- **안전 순서열** : 교착 상태 없이 안전하게 프로세스들에 자원을 할당할 수 있는 순서를 의미

- 안전 순서열이 존재: $P2 \rightarrow P1 \rightarrow P3$

프로세스	요구량	현재 사용량
P1	10	5
P2	4	2
P3	9	2

- 할당 가능 자원: **12**
- 할당한 자원(P1, P2, P3 현재 사용량의 총합): **9**
- 남은 자원(할당 가능 자원 - 할당한 자원): **3**

- 만일 아래의 상황에서 운영체제가 P3에게 선뜻 자원 하나를 해주었다면?

프로세스	최대 요구량	현재 사용량
P1	10	5
P2	4	2
P3	9	3

- 할당 가능 자원: **12**
- 할당한 자원(P1, P2, P3 현재 사용량의 총합): **10**
- 남은 자원(할당 가능 자원 - 할당한 자원): **2**

- P2 먼저 처리해도 남은 자원은 4이고, 해당 자원으로는 P1과 P3를 처리할 수 없음
- 운영체제가 교착 상태를 회피하기 위해서는 시스템 상태가 안전 상태에서 안전 상태로 움직이는 경우에만 자원을 할당하면 됨

교착 상태 검출 후 회복

- 교착 상태 발생을 인정하고 사후에 조치하는 방식
- 선점을 통한 회복
 - 교착 상태가 해결될 때까지 한 프로세스씩 자원을 몰아주는 방식
 - 많은 프로세스들이 작업 내역을 일게 될 가능성이 있음
- 프로세스 강제 종료를 통한 회복
 - 교착 상태가 없어질 때까지 한 프로세스씩 강제 종료
 - 교착 상태가 없어졌는지 여부를 확인하는 과정에서 오버헤드를 야기