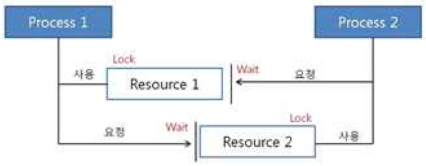
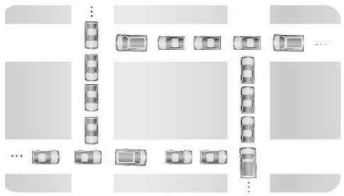
Dead Lock (교차상태)

데드락은 상호 배제에 의해 나타나는 문제점이다.
둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 무한정 기다리는 현상을 의미.

프로세스가 자원을 얻지 못해 다음 처리를 하지 못하는 상태로, 시스템적으로 한정된 자원을 여러곳에서 사용하려고 할 때 주로 발생





p1과 p2가 리소스 a,b를 둘다 얻어야할 때, p1이 리소스 a를 얻고, p2는 리소스 b를 얻는다.
그러나, 서로가 가진 리소스를 기다려야한다.
하지만 두 프로세스는 원하는 리소스가 상대방에게 할당되어 있어, 무한정 기다리게 된다.

보통 1. 한정된 자원을 사용하려고 할 때 경쟁을 하고
2. 어떤 프로세스가 자원을 요청했을 때 그 시각에 그 자원을 사용할 수 없는 상황 때
3. 대기 상태로 들어간 프로세스들이 실행 상태로 변경 될 수 없을 때.

-필요 충분 조건-
1) 상호배제 : 한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 합니다.
2) 점유와 대기 : 최소한 하나의 자원을 점유 + 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야한다.
3) 비선점 : 다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 뺏을 수 없다.
4) 순환대기 : 서로가 가지고 있는 자원을 얻기 위해 대기하는 상태.

{상점순비} 한 개의 프로세스는 한 개 자원, 하나 점유하고 다른 자원 점유 대기, 뺏기 않기, 서로 대기

예방 및 회피 기법

<예방기법> 교착 상태 발생 조건 중 하나를 제거함으로써 해결하는 방법 (자원 낭비가 심함)

1) 상호배제 부정 : 여러 프로세스가 공유 자원을 사용할 수 있게 한다.
2) 점유 대기 부정 : 프로세스가 실행되기 전에 필요한 모든 자원을 할당한다.
3) 비선점 부정 : 자원을 점유하고 있는 프로세스가 다른 자원 요구 시 자원을 반납하고 대기
4) 순환 대기 부정 : 자원에 고유한 번호를 할당하고 번호 순서대로 자원을 요구하도록 함.

<회피기법> 교착상태가 발생하면 피해나가는 방법

은행원 알고리즘 (다익스트라가 제안한 방법)
1. 은행에서 모든 고객의 요구가 충족되도록 현금을 할당하는 데서 유래
2. 각 프로세스가 자원을 요구할 때, 시스템은 자원을 할당한 후에도 안정 상태로 남아있는지 사전에 검사하여 교착상태를 회피함.
3. 안정상태에 있다면, 자원을 할당하고 그렇지 않다면 자원 해지까지 대기함.
4. 교착 상태가 되지 않도록 보장하기 위해, 교착 상태를 예방하거나, 회피하는 프로토콜 이용

<회복기법>
교착상태를 일으킨 프로세스를 종료하거나 교착상태에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 기법

1) 프로세스를 종료하는 방법 2가지
   1. 교착상태의 프로세스를 모두 중지
   2. 교착 상태가 제거될 때까지 한 프로세스씩 중지.

2) 자원을 선점하는 방법
   1. 교착 상태의 프로세스가 점유하고 있는 자원을 선점하여 다른 프로세스에게 할당.
      -> 이때 해당 프로세스는 일시 정지 시킨다.
   2. **우선순위**가 낮은 프로세스, 수행된 횟수가 적은 프로세스 등을 위주로 프로세스의 자원을 선점한다.

<탐지 기법>
교착상태 탐지는 시스템 교착상태가 있는지 점검하고, 있다면 탐지.
하지만 자원을 요청할 때마다 탐지 알고리즘을 실행하면 그에 대한 오버헤드가 생긴다.