**OS 중간고사 정리**

시험 출제

Chapter 7. Deadlocks

P0 P1

wait(A) wait(B)

wait(B) wait(A)

여기서 인터럽트 발생시 프로세스가 모두 waiting상태에 빠져서 못 나옴. → deadlock

**데드락의 필요조건**

데드락이 발생되면, 4가지 조건이 만족된다.

1. Mutual exclusion: 한 사람만 사용 가능
2. Hold and wait: semaphore를 하나 갖고 있으면서 하나 더 요구할 때
3. No preemption: 중단하지 않음 (양보X)
4. Circular wait: 원형으로 기다림

**데드락의 해결방법**

1. Prevention: 데드락이 안 일어나게 프로그래밍 → 유일한 방법
2. Avoidance: OS가 데드락에 빠질 위험이 있으면 할당을 하지 않는다.
3. Detection and recovery: OS가 계속 데드락을 검사하고 데드락에 빠지면 회복시킨다.

회복: 이전 상태로 돌아가거나 다 죽이고 새로 시작.

* ②, ③에서 데드락인지 검사하는 과정이 오버헤드가 크다.
* 다 죽이고 새로 시작하는 회복은 좋은 방법이 아니다.

**데드락 예방방법**

4가지 조건을 전부 만족시키지 않으면, 데드락이 일어나지 않는다. (대우)

1. Mutual exclusion: 안 쓸 수는 없다.
2. Hold and wait: 고려하기 쉽다. ex) 밥 먹는 철학자에서 semaphore 하나로 변경
3. No preemption: 코드 짜기 힘들다.
4. Circular wait: 일자인 경우 데드락 발생X

* ②, ④을 고려하기 쉽다.