14. Semaphores

Critical Section

- entry section
  request permission
- critical section
  shared resource, non-re-entrant code
- exit section
  release permission
- reminder section
  그 이후 동작.

Solution for critical section

- Mutual Exclusion
  critical section에 한번에 하나만.
- Progress
  다중 프로세스를 계속 critical section에 진입시킬 수 있어야 한다.
- Bounded Waiting
  기다리는 시간에 제한이 있어야 한다. 제한시간 안에는 critical section에 들어가야 한다.

Semaphores

wait: 0보다 크면 1 감소시킨다. / 0이면 block 된다.

Signal: 0이고 block되어있는게 없으면 깨운다 / block된 것이 없으면 1 증가시킨다.

- Critical section w. semaphores

초기값 = 1 (초기값이 0이면, wait에서 dead lock)

wait (초기값이 8이면, 최대 8개가 동시에 접근 가능)

<critical section>

signal

S, Q 초기값 = 1

wait S	wait Q	하나이상 차이날 수 없게 제한함.
a	b	
signal Q	signal S	(초기값이 1,0 이면 무조건 한 갈아가면서 함)
		(초기값이 0,0 이면 dead lock)

- init, destroy
  - sem_init (sem, pshared, value)
  - sem_destroy (sem)

- Operations
  - sem_post (sem) = signal 역할
  - sem_wait (sem)
  - sem_trywait (sem) 값이 0이면서 못 들어가면 바로 -1 리턴
  - sem_getvalue (sem, value) 그때 당시의 세마포어 값인지는 알 수 없음.

Named Semaphores

이름 / 프로세스에게 전달

Create, open

sem_open (name, o_flag, permission (0___ 형식), 세마포어 초기값)

↓

O_CREATE 생성, 같은 이름의 세마포어 있으면 open

O_CREATE | O_EXCL 이미 존재하면 오류

0 이면 그 이름의 기존 세마포어 open