

동기화 문제 해결 보고서

문제: 흑석동에 있는 주요 교차로를 통과하는 교통량은 지난 몇 년 동안 증가했습니다. 이제 막대한 교착 상태로 인해 중앙대학교로 등교하거나 출근하는 사람은 교통이 교차로를 통과하는 데보다 효율적인 방법이 필요하다는 사실을 인정하게 되었습니다. 귀하의 임무는 세마포어를 사용하여 솔루션을 설계하고 구현하는 것입니다. 구현을 해서, 교착상태를 어떻게 해결하는 지를 설명하는 보고서를 작성해서, 제출하세요.

1. 직진 : NW-SW

우리는 4차선의 교차로에서 각 자동차를 출발시켜 이 교차로상에 동기화문제가 발생하지 않도록 조작하여야 합니다.

나는 일단 4가지 진행방향을 dir이라 명명하고 정리하였습니다.

dir : 0

in -> nw -> sw -> out

dir : 1

in -> ne -> nw -> out

dir : 2

in -> se -> ne -> out

dir : 3

in -> sw -> se -> out

각 차량의 방향은 (차량의 번호)%4로 설정하였다.

또한 각 이동은 다음과 같은 순서로 이어집니다

1번칸의 점유

Print(1번칸을 점유했습니다.)

2번칸의 점유

Print(1번칸에서 2번칸으로 이동 중)

1번칸 탈출

Print(2번칸을 점유중입니다.)

2번칸 탈출

Print(차량이 교차로를 통과했습니다.)

데드락 문제를 해결하기위해 제가 선택한 방법은 Prevention , Avoidance , Recovery 중 Avoidance 를 이용하기로 하였습니다.

Prevention: 데드락이 일어나지 않도록 신호등을 설치하는 방식을 생각하였으나 이는 시간이 오래 걸릴 것으로 예상하였습니다.

Recovery : Recovery 이용하려면 자동차의 후진을 구현해야 하는데 그것은 문제해결을 보다 유연 하게 할 수 있도록 해주지만 복잡성 때문에 시간이 오래걸릴것으로 예상하였습니다.

따라서 저는 교차로에 차량이 3대 존재할 경우 더 이상 차량이 진입할 수 없도록 하여 새로운자원(차량진입)이 일어날 때 조건을 확인하여 데드락을 피해보았습니다.

따라서 새로이 추가해줘야 할 semaphore가 생겼습니다.

Testsem: 진입한 차량의 수를 조정해 줄 semaphore, semcount 초기화를 조정로하여 차량이 최대 3개까지 진입할 수 있도록 프로그래밍 하였습니다.

nwsem, nesem, swsem, sesem: 교차로의 4방향을 의미하는 semaphore 자동차가 이를 할당하면 sem_count를 낮춰 다른 자동차가 진입할 수 없도록 합니다.

Printsem : 현재 상태를 print해주기 위해 이용할 semaphore (출력 겹침해결)

Donesem: 차량이 진입이 끝나면 다음진입을 위해 필요한 semaphore

각 함수의 변경 및 작성사항은 보고서로 작성하였습니다.

Inititems 함수

각 교차로 세마포어와 프린트 세마포어를 추가하여 생성하였고 카운트를 1로 시작하여 1번에 1개의 자원만 할당할 수 있도록 하였다.

go_straight 함수

thread에 적용될 함수 testsem을 이용하여 최대 3개의 차량만 교차로에 존재할 수 있다.

위에서 설명한 이동방식을 semaphore를 이용해 적용한다.

Semtest 함수

Thread에 적용되는 함수를 gostraight로 변경

실행화면 :

```
car num 27 is completely out
car num 29 is come to ne semaphore
car num 28 is come to nw semaphore
car num 28 is out from nw and go to sw semaphore
car num 28 is come to sw semaphore
car num 28 is completely out
car num 29 is out from ne and go to nw semaphore
car num 30 is come to se semaphore
car num 30 is out from se and go to ne semaphore
car num 30 is come to ne semaphore
car num 30 is completely out
car num 29 is come to nw semaphore
car num 29 is completely out
car num 31 is come to sw semaphore
car num 31 is out from sw and go to se semaphore
car num 31 is come to se semaphore
car num 31 is completely out
car num 8 is come to nw semaphore
car num 8 is out from nw and go to sw semaphore
car num 8 is come to sw semaphore
car num 8 is completely out
Semaphore test done.
Operation took 9.417836440 seconds
OS/161 kernel [? for menu]:
```

실행화면에 대한 설명

교차로에 차량이 4대를 넘어가지 않아 deadlock이 발생하지 않는다.

만약 testsem을 조작해 교차로에 차량이 4대까지 존재할 수 있도록 프로그램을 작성하면?

실행화면:

```
OS/161 kernel [? for menu]: sy1
Starting semaphore test...
If this hangs, it's broken:
car num 0 is come to nw semaphore
car num 1 is come to ne semaphore
car num 2 is come to se semaphore
car num 3 is come to sw semaphore
```

0,1,2,3 차량이 들어오며 4방향의 모든세마포어를 선점하고 서로가 서로의 교차로로 진입하려는 deadlock이 발생합니다.