

운영체제

-mylock 구현-

소프트웨어학과

201520860

노근탁

구현 과정

-**spinlock** : spinlock의 경우 주어진 `compare_and_swap`을 이용해서 atomic하게 turn 값을 변경하여 스레드가 lock을 잡고있으면 while문을 계속 시행하고 lock을 release 하면 turn 값을 변경하여 while 문을 돌고있는 스레드를 탈출시켜 lock을 잡을 수 있게하였습니다. Ring buffer는 배열기반으로 만들었고 head와 tail값을 가지고 연산하여 버퍼가 꽉찼거나 비었으면 goto문을 이용하여 overflow underflow 문제를 해결하였습니다.

-**mutex** : 이번 과제에서 구현해야하는 뮤텍스는 blocking mutex이기 때문에 value를 1로 초기화하고 스레드가 락을 잡고있으면 pause로 재우고 release 하면 시그널을 보내 waiting list에서 자고있는 스레드를 깨워서 락을 잡게 하였습니다. 뮤텍스 과정에서 value값을 바꾸는 과정이 atomic 하지않게 실행될수있기 때문에 spinlock을 이용해서 이를 해결하였습니다. Mutex enqueue , dequeue 도 spinlock의 경우와 마찬가지로 방법으로 overflow underflow문제를 해결하였습니다.

-**semaphore** : semaphore의 경우 lock하는 코드는 mutex에서 구현한 것과 대부분 동일했고 lock을 초기화할 때 value값을 empty의 경우는 slot개수만큼 full의 경우 0로 mutex의 경우 1로 초기화하는 부분만 다르게 구현하였습니다. Enqueue dequeue 부분은 semaphore empty 와 semaphore mutex semaphore full 을 조합하여서 counting semaphore로 ring buffer를 구현하였습니다.

고찰 ∴ 스핀락은 수업시간에 들은 내용으로 구현하고 테스트락도 assert의내용을 적당히 바꿔서 제대로 작동하는 것을 확인하였으나 뮤텍스구현에서 list_head 사용방법과 시점에 대해서 감을 못 잡고있었다가 교수님의 유튜브 라이브를 통해 힌트를 얻어 해결하였습니다. 세마포어의 경우 뮤텍스와 개념적으로 이해를 하지 못하다가 강의자료의 bounded buffer 문제 해결하는 예시를 하나씩 논리적으로 코드진행을 생각해보니 mutex와의 차이와 구현방법에 대해 이해할 수 있었습니다.

확실히 이번학기 개념적인 수업보다 힘들지만 직접 구현하며 더욱 직접적으로 이해할 수 있는 기회가 있어서 좋은 것 같습니다. 추가로 저번 과제였던 Advmysh2 구현하는 방법 혹은 구현된 코드진행도 공유해주시면 비록 미완성인 코드로 냈지만 추후에 도움이 될 수 있을 것 같아 부탁드립니다.