

Starve-Free Reader Writer With Zemapaphore

제출자 : 20191564 김신건

Github: <https://github.com/shinkeonkim/starve-free-reader-writer-with-zemaphore>

문제 정의

Reader-Writer 문제는 여러 Thread가 동시에 접근할 때 발생하는 Concurrency 문제입니다.

이 문제에서:

- Writer는 한 번에 하나만 critical section에 접근할 수 있습니다.
- Reader는 여러 개가 동시에 critical section에 접근할 수 있습니다.
- Writer와 Reader는 서로 동시에 critical section에 접근할 수 없습니다.

Starve-Free 해결책

기존의 Reader-Writer 문제 해결책에서는 Reader가 계속해서 들어오는 경우 Writer가 무한정 기다리는 starvation 문제가 발생할 수 있습니다. 이를 해결하기 위해 다음과 같은 방법을 구현했습니다:

사용된 세마포어

read_try: Writer가 기다리고 있는 경우, Reader를 Block하기 위한 세마포어

구현된 함수들

초기화

```
void rwlock_init(rwlock_t *rw) {
    rw->readers = 0;
    rw->waiting_writers = 0; // ADDED LINE
    sem_init(&rw->lock, 0, 1);
    sem_init(&rw->writelock, 0, 1);
    sem_init(&rw->read_try, 0, 1); // ADDED LINE
}
```

Reader Lock

```
void rwlock_acquire_readlock(rwlock_t *rw) {
    sem_wait(&rw->read_try); // ADDED LINE
    sem_wait(&rw->lock);
    rw->readers++;
    if (rw->readers == 1)
        sem_wait(&rw->writelock);
    sem_post(&rw->lock);
}
```

```
sem_post(&rw->read_try); // ADDED LINE
}
```

Writer Lock

```
void rwlock_acquire_writelock(rwlock_t *rw) {
    sem_wait(&rw->read_try); // ADDED LINE
    rw->waiting_writers++; // ADDED LINE
    sem_wait(&rw->writelock);
    rw->waiting_writers--; // ADDED LINE
}
```

Writer Unlock

```
void rwlock_release_writelock(rwlock_t *rw) {
    sem_post(&rw->writelock);
    sem_post(&rw->read_try); // ADDED LINE
}
```

실행 과정

1. 초기화

- 모든 세마포어를 1로 초기화
- readers와 waiting_writers 카운트를 0으로 초기화

2. Reader 동작

- **read_try** 획득
- **lock** 획득하여 readers 수 증가
- 첫 번째 reader인 경우 **writelock** 획득
- **lock** 해제
- **read_try** 해제
- 작업 완료 후 **lock** 획득하여 readers 수 감소
- 마지막 reader인 경우 **writelock** 해제
- **lock** 해제

3. Writer 동작

- **read_try** 획득하여 새로운 reader 진입 방지
- waiting_writers 카운트 증가
- **writelock** 획득
- waiting_writers 카운트 감소
- 작업 완료 후 **writelock**과 **read_try** 해제

Starvation 방지 메커니즘

1. read_try의 역할

- Writer가 `read_try`를 획득하면 새로운 Reader가 진입할 수 없음
- Writer가 작업을 마치면 `read_try`를 해제하여 Reader가 다시 진입 가능
- 이를 통해 Writer가 무한정 기다리는 상황 방지

2. waiting_writers 카운트

- Writer가 진입할 때 증가, 진입 후 감소

컴파일 및 실행

```
> make
```

실행 결과 예시

```
> make
g++ -std=c++11 -pthread -o main main.cpp -g -Wall && ./main && rm -f main
Reader 0: reading...
Reader 1: reading...
Reader 0: done.
Reader 2: reading...
Reader 1: done.
Reader 3: reading...
Reader 2: done.
Reader 4: reading...
Reader 3: done.
Reader 5: reading...
Reader 4: done.
Reader 5: done.
Writer 1: writing...
Writer 1: done.
Reader 6: reading...
Reader 7: reading...
Reader 8: reading...
Reader 9: reading...
Reader 6: done.
Reader 7: done.
Reader 8: done.
Reader 9: done.
```