

Operating Systems

Project #1

소프트웨어학부

2020년 5월 21일

Reader-Writer 문제

Reader와 writer가 공유자원에 접근할 때, reader는 다른 reader와 동시에 접근할 수 있지만, writer가 다른 writer나 reader와 동시에 접근하면 문제가 발생할 수 있는 것을 reader-writer 문제라고 한다. 이 문제를 해결하기 위한 해법에는 reader 선호, writer 선호, 공정한 reader-writer, 세 가지가 있다. 어떤 방식이든 모두 writer에게는 상호배타를 보장하고, reader에게는 다른 reader와 최대한 중복을 허용하는 것을 목표로 한다.

Reader 선호

수업시간에 배운 해법으로 CS (Critical Section)에 reader가 있으면 다른 reader가 중복해서 들어갈 수 있다. 그러나 이 방식은 늦게 온 reader가 기다리고 있는 writer를 앞지를 수 있어서 writer가 굼주릴 수 있다. 특히 reader의 수가 상대적으로 많은 환경에서는 writer가 기회조차 얻지 못하는 경우가 발생할지도 모른다.

Writer 선호

Reader의 중복을 최대한 허용하되 기다리는 writer가 있으면 reader가 더 이상 writer를 앞지르지 못하게 하는 방식이다. 뿐만 아니라 늦게 온 writer는 기다리고 있는 reader를 앞지를 수 있다. 이러한 방식을 writer 선호라 부르고, writer의 수가 상대적으로 적은 환경에서 유리하다.

공정한 reader-writer

Writer선호 방식은 반대로 reader가 굼주릴 수 있다. 공정한 reader-writer는 선착순으로 CS에 들어가면서 reader의 중복을 최대한 허용하는 방식이다. 이 방식에서는 늦게 온 reader/writer가 기다리고 있는 다른 reader/writer를 앞지르지 못한다.

과제 요구사항

시스템에 10개의 reader 스레드와 2개의 writer 스레드가 있다. 각 스레드는 CS에 들어가서 필요한 공유자원을 읽거나 쓴다. 12개의 스레드가 올바르게 동기화하며 동작할 수 있도록 앞에서 언급한 세 가지 방식을 사용하여 reader-writer 문제를 푼다. 스레드가 CS에 들어가서 공유자원을 접근하는 부분은 코딩에서 생략한다. 다만 스레드가 CS에 들어왔다는 것을 보이기 위해 다음과 같은 일을 반복해서 수행한다.

- Reader가 CS에 들어가면 첫 번째 reader는 A 문자를, 두 번째 reader는 B 문자를, ..., 이런 식으로 각 reader는 같은 문자를 8192개 출력하고 CS를 빠져나온다.
- Writer가 CS에 들어가면 어떤 배열에 저장된 문자를 출력한다. 출력이 끝나면 CS를 빠져나와 0.1초 잠잔다.

앞에서 요구한 reader와 writer는 `skeleton.c`라는 골격파일에 이미 구현이 된 상태로 학생들에게 제공된다. 다만 CS 접근을 통제할 수 있는 스레드 동기화가 빠져있다. 이 골격파일을 사용하여 reader-writer 문제를 푸는 세 가지 방식을 설계하고, POSIX pthread API를 사용하여 구현한다. Reader 선호 해법은 수업시간에 배운 것을 그대로 사용하면 된다. 다만 이 경우 reader가 기다리는 writer를 계속 앞지르기 때문에 writer가 전혀 기회를 못 얻는 것이 정상이다.

동기화 도구

학생들은 POSIX의 mutex lock이나 세마포를 사용해서 세 가지 해법을 모두 구현할 수 있다. 코딩해야 하는 양이 많지는 않지만 병행 프로그램의 특성상 오류를 범하기 쉽고, 디버깅하기가 다소 까다롭다. 어쩌면 생각했던 것보다 시간이 더 걸릴지도 모른다. 만일 조건변수를 적절히 사용한다면 비교적 이해하기 쉽고 간결하게 코딩을 마칠 수도 있다. 다행히 POSIX pthread는 모니터를 제공하지는 않지만 조건변수를 제공한다. 특히 writer 선호 해법에는 조건변수를 사용할 것을 권장한다. 물론 시간적 여유가 있으면 다른 방법을 경험해보는 것도 좋을 것이다.

skeleton.c

앞에서 언급한 것처럼 골격파일에는 reader와 writer가 이미 구현되어 있다. 이 골격파일을 컴파일해서 실행하면 약 0.5초 동안 12개의 스레드가 동기화 없이 각자 출력을 쏟아내고 종료한다. 컴퓨터 성능에 따라 차이가 있겠지만 12개의 스레드가 수 MB가량 문자를 출력한다. 이 골격파일은 macOS에서 검증되었다. 하지만 Ubuntu나 다른 Linux 환경에서도 문제없이 잘 돌아갈 것이다.

제출물

Reader-writer 문제에 대한 세 가지 해법이 잘 설계되고 구현되었다는 것을 보여주는 자료를 각자가 판단하여 PDF로 묶어서 이름_학번_PROJ1.pdf로 제출한다. 여기에는 다음과 같은 것이 반드시 포함되어야 한다.

- 본인이 설계한 writer 선호, 공정한 reader-writer 알고리즘 (1쪽 분량)
- 프로그램 소스파일 3개
- 컴파일 과정을 보여주는 화면 캡처
- 실행 결과물의 주요 장면과 그에 대한 설명 및 차이점 등

HK