

Assignment3-Mylock Report

- 소프트웨어학과 201520882 천윤서

1. 무엇을 했고 어디까지 했는지.

Spinlock을 구현하여, 여러 스레드들이 Circular Queue에 안전하게 접근할 수 있도록 하였고, 이 Spinlock과 Linux LIST_HEAD의 3개의 자료구조 중 하나인 TAILQ를 이용하여 스레드가 대기할 수 있는 큐를 구현하여서 둘을 통해 Mutex를 구현하였다. 또, 이를 변형하여 리소스가 여러 개 일 때 쓸 수 있는 Semaphore를 구현하였다. 단, 이 과제에서 Semaphore는 초기화 시 value가 1로 초기화 되어서 Binary Semaphore이다.

2. 어떤 것을 배웠는지.

Spinlock을 구현하여, 직접 Queue를 race-condition을 없애려면 어디부터 어디까지 제어해야 하는지 살펴보았다. Mutex와 Semaphore를 구현했는데, Semaphore가 Mutex의 단순 응용형이라 별 차이가 없다는 것을 쉽게 알았다. 또한, 어디서부터 어디까지 보호하는가에 따라 성능이 천차만별인 것을 알았다.

3. 과제에 대한 피드백.

이미 제시된 자료구조인 LIST_HEAD를 통해서 온전히 Lock 구현에 힘을 쏟을 수 있었다. 다만 처음에 Generator가 너무 많아서 한 번 테스트하는데 5분씩 걸렸고, 이는 토요일에 코드가 수정되면서 해결됐다. 구 Generator가 테스트하는데 5분이 걸렸다면, 수정된 코드의 Generator는 약 20초 정도가 걸리기에 수정된 쪽이 더 학습에는 적합하다고 본다.(테스트 하는 동안 CPU 점유율이 치솟아서 다른 공부를 하기 힘들어서 쉬고 왔다.)