시스템 프로그래밍(화요일)

최상호 교수님

Proxy 3-1

컴퓨터정보공학부

2018202076

이연걸

- Introduction

  이번 과제는 logfile에 관한 동시접근을 제어하는 과제이다. 3개의 프로세스가 현재 값이 10인 a를 1증가시켜서 13을 만들기 위한 과정을 보자

  프로세스 1은 a의 값을 읽어온다. a=10

  프로세스 2는 a의 값을 읽어온다. a=10

  프로세스 1은 a의 값을 증가시킨다. a=11

  프로세스 3은 a의 값을 읽어온다. a=11

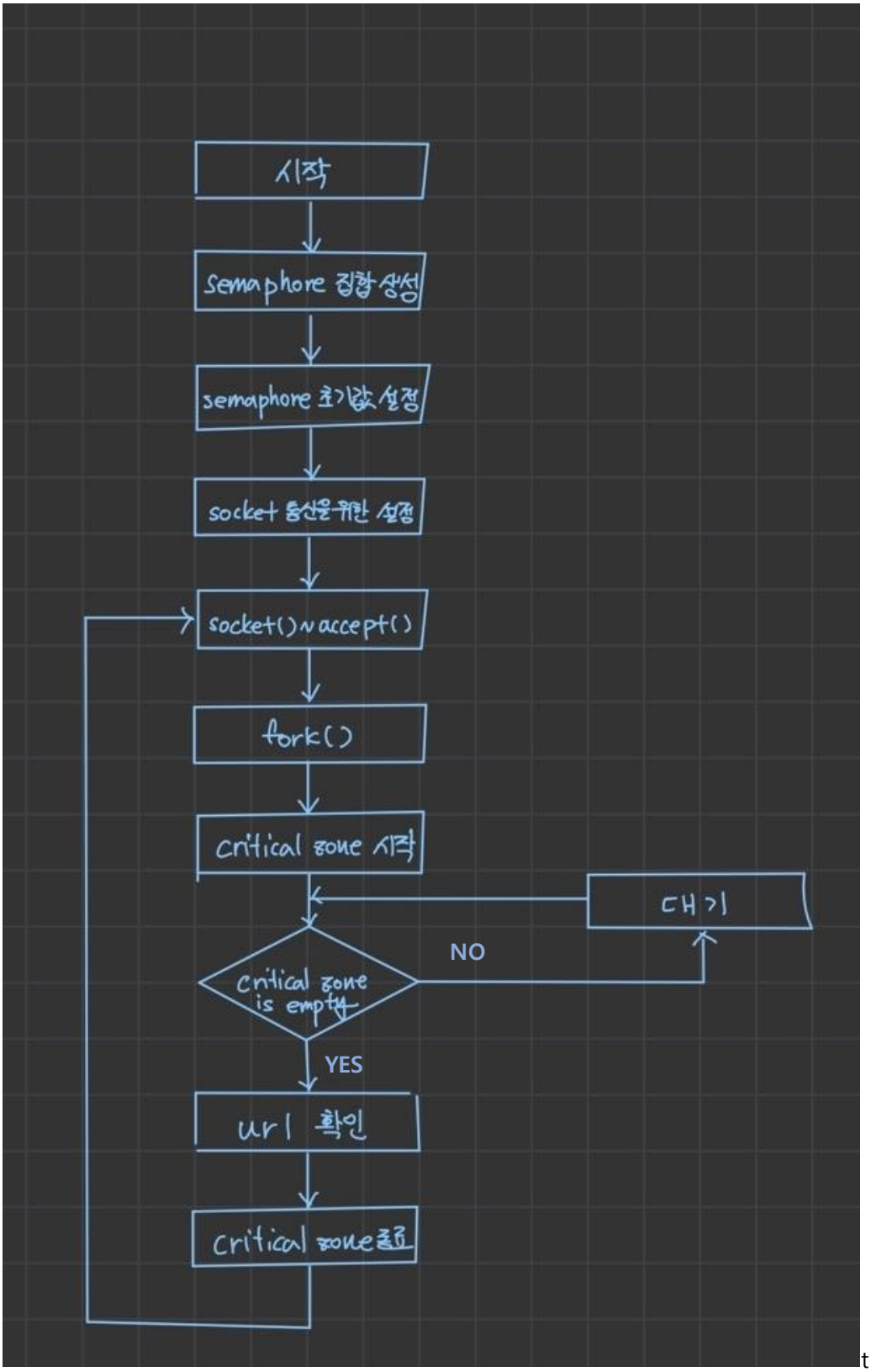
  프로세스 2는 a의 값을 증가시킨다. a=11

  프로세스 3은 a의 값을 증가시킨다. a=12

  이렇게 내가 예상한 값과 전혀 다른 값이 나오게 된다. 이때 a를 Critical Zone이라고 한다. 여러 개의 프로세스가 Critical Zone에 동시에 접근했기 때문에 위처럼 잘못된 결과가 나온 것이고 이를 막기 위해서 Semaphore를 사용한다. 따라서 Semephore를 사용한다는 말은 동시접근을 제어한다는 말과 같다. Semaphore는 P, V연산으로 이루어져 있는데 flag가 1일때만 Critical Zone에 접근이 가능하다고 가정하자. P연산은 flag를 1감소, V연산은 flag를 1증가시킨다. 이 두개의 연산을 반복하며 flag가 1일때만 Critical Zone에 접근이 가능하게 만든다면 한 개의 프로세스만 공유자원 즉 Critical Zone에 접근할 수 있다. 물론 flag가 2, 3 … 이 되어서 특정 개수의 프로세스만 공유자원에 접근 할 수 있게 수정할 수 있다.

  이를 활용하여 여러 개의 프로세스 중 한 개의 프로세스만 공유자원 즉 Logfile.txt에 접근해 작업을 할 수 있게 만든다.

- Flow Chart



t

- Pseudo Code

  시작

Semaphore 집합 생성

Semaphore 초기값 설정

Socket통신용 옵션들 설정

Socket()

Bind()

Listen()

while

Accept()

Fork()

Critical Zone시작

If Critical Zone is empty Then continue

Else

대기

로그, 캐시 파일 기록

Critical Zone종료

End While

- 시뮬레이션 시나리오

프로세스 1이 Critical Zone에 접근

프로세스 2가 Critical Zone에 접근 -> 거부 -> 대기

프로세스 1이 Critical Zone에 위치 -> 작업중

프로세스 3이 Critical Zone에 접근 -> 거부 -> 대기

프로세스 4가 Critical Zone에 접근 -> 거부 -> 대기

프로세스 1 작업 끝 -> Critical Zone에서 탈출

프로세스 2가 Critical Zone에 위치 -> 작업중

프로세스 5가 Critical Zone에 접근 -> 거부

프로세스 2 작업 끝 -> Critical Zone에서 탈출

프로세스 3이 Critical Zone에 위치 -> 작업중

프로세스 3 작업 끝 -> Critical Zone에서 탈출

프로세스 4가 Critical Zone에 위치 -> 작업중

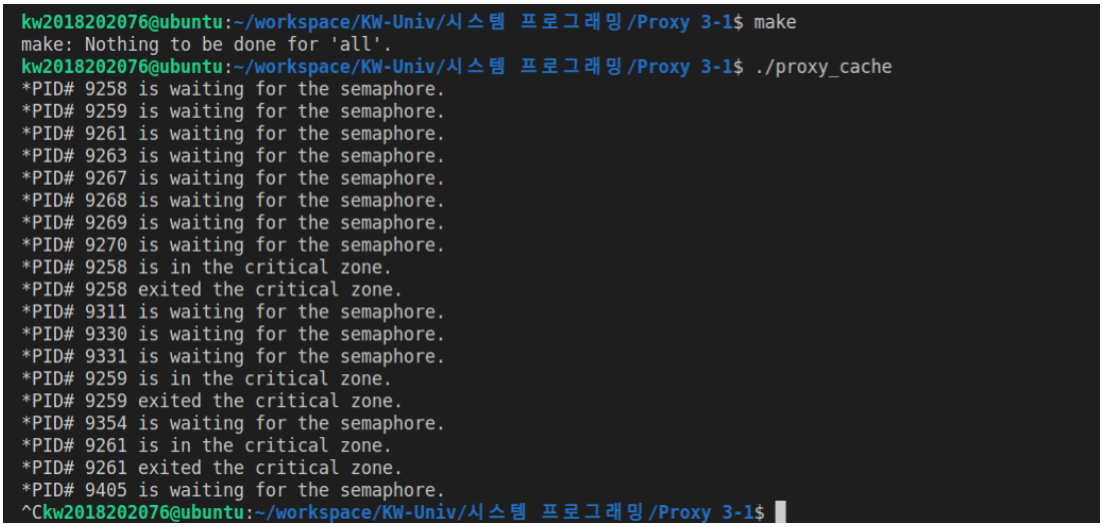
프로세스 4 작업 끝 -> Critical Zone에서 탈출

프로세스 5가 Critical Zone에 위치 -> 작업중

프로세스 5 작업 끝 -> Critical Zone에서 탈출

프로세스 5개로 만든 시뮬레이션

- 결과 화면



- 고찰

이번 과제는 Semaphore의 이해도가 중요했다. 이렇게 말한 이유는 처음에 단순이 다수의 프로세스가 접근하는 내용을 printf 함수를 출력하는 줄 알았지만 Semaphore는 공유자원에 대한 접근을 관리하는 것이므로 공유자원이 어디인지를 알아야 했다.

문제에서 주어진 조건으로는 한 개의 프로세스만 logfile에 접근할 수 있다고 명시했으므로 URL을 확인해 Logfile에 Write하는 부분이 공유 자원(Critical Zone)에 해당한다.

Semephore를 사용하기 위해서는 여러 시스템 콜을 사용해 Semaphore 집합 생성, 초기값 설정을 해주어야 했다. 이번에는 강의 자료에서 명시가 되어있어서 쉽게 할 수 있었지만 명시되지 않았더라면 시간이 오래 걸렸을 것이다.

이번 과제에서 공유 자원(resource)에 접근 할 수 있는 최대 프로세스 개수를 1로 설정했는데 이는 Mutex의 기능과 동일하다. Mutex는 오직 1개의 프로세스 혹은 스레드만이 Critical Zone에 접근 할 수 있고, Semaphore는 지정된 변수 semget() 시스템 콜의 int nsems 의 값만큼 접근 할 수 있다.