

2023년 2학기 스터디그룹 주간학습보고서

그룹 명	MaveOS		
날 짜	11월 15일	시 간	12시 00분 ~ 14시 45분
스터디장소	50주년 기념관 516호	회차	9 회차 모임
수강과목	운영체제	담당교수	강지훈
스터디	리더: 한아림		
참석자	그룹원: 손효림, 이경화		

학습문제

<동기화>

- 임계구역 문제
- Peterson 해결안
- Semaphores

학습문제 해결과정

<동기화>

#1. 임계구역 문제

1) 임계 구역(Critical Section)

- N개의 프로세스가 있는 상황에서 각 프로세스가 포함하고 있는 코드 부분
- 임계구역에서는 공유 데이터에 접근하고 갱신할 수 있음
- 임계구역에는 한 번에 하나의 프로세스만 접근 및 수행할 수 있음

2) 임계구역 문제

- 상호 배제(Mutual exclusion) : 하나의 프로세스가 임계구역에서 실행되면 다른 프로세스들은 자신의 임계구역에서 실행될 수 없음
- 진행(Progress) : 임계구역에서 실행되는 프로세스가 없을 때 프로세스 중 누가 먼저 들어갈 것인가에 대한 결정이 유한 시간 내에 일어나야 함
- 한정된 대기(Bounded waiting) : 프로세스가 임계구역에 진입하려는 요청을 한 후부터 요청이 허용될 때까지 유한 시간 내에 임계구역 안에 들어갈 수 있어야 함

#2. Peterson 해결안

1) Peterson 해결안

- 두 프로세스 P1과 P2가 2개의 데이터를 공유하여 임계구역 문제 해결
- int turn : 임계구역에 들어갈 차례
- boolean flag[2] : 임계구역으로 진입할 준비가 되었는지 여부
- 프로세스들은 자신이 임계구역에 들어갈 준비가 되면 flag값을 true로 입력하고 turn 값을 상대방으로 지정함

2) Peterson 해결안 증명

- 상호 배제(Mutual exclusion) : 상호 배제가 제대로 지켜지는지
 - flag가 true이고 turn이 자신인 프로세스가 실행되고 다른 프로세스는 막힘
- 진행(Progress) : 진행에 대한 요구 조건을 만족하는지
 - 동시에 i와 j가 입력되어도 결국 하나의 값만 저장되므로 누가 먼저 임계구역에 들어갈 것인지 유한 시간 내에 결정됨
- 한정된 대기(Bounded waiting) : 대기 시간이 한없이 길어지지 않는지
 - 임계구역 작업이 끝났을 때 자신의 flag를 false로 바꿔서 대기 중인 프로세스가 임계구역에 접근할 수 있도록 함

2) Peterson 해결안의 한계

- 최신 컴퓨터 아키텍처에서 사용할 수 없음
- 주된 이유는 프로세서 또는 컴파일러가 종속성 없는 읽기 및 쓰기 작업에 대한 순서를 재정렬하기 때문
- turn 값과 flag 값의 순서 재정렬로 인해 두 개의 프로세스가 동시에 임계영역에 접근하는 문제가 발생할 수 있음

#3. Semaphores

1) Semaphores


- Semaphore란 : 동기화 대상이 1개 이상일 때 사용하는 동기화 도구
- 작동 방식 : 두 개의 원자적 함수로 제어되는 정수 변수를 통해 공유 자원에 대한 접근을 제어
- 구분 : 카운팅 semaphore, 이진 semaphore

2) 카운팅 Semaphore 사용법

- synch 변수 : 두 프로세스가 공유함. 0으로 초기화. 자원을 사용하고자 하는 프로세스는 synch 값이 0보다 커질 때까지 block됨.
- wait(synch) 함수 : synch 값 1 감소
- signal(synch) 함수 : synch 값 1 증가

3) Semaphore 구현

- 아직까지 mutex와 semaphore는 busy waiting을 사용.
- busy waiting 대신에 프로세스가 자기 자신을 일시중지하는 방법을 사용할 수 있음

학습성찰	
학습내용 이해도	98 %
<p>학습활동 돌아보기</p> <p>(좋았던 점, 아쉬운 점 활동모습 사진 추가)</p>	<p>- 한아림 이번 주차 스터디에서는 저번에 학습하였던 내용에 이어 count ++ 와 count --를 간단한 기계어로 구현해보는 실습부터 진행하였다. 단 하나의 CPU에서만 접근할 수 있는 로컬 레지스터를 추가적으로 할당하여 메모리 값 이동의 흐름까지 파악할 수 있었다. 이 뿐만 아니라 버퍼의 동작 과정, producer & consumer 의 레지스터 접근 방식에 대해서도 보다 더 가시적으로 이해할 수 있었다. 또 저번 시간에 교수님께서 강조하신 Semaphore 사용법에 대해서도 공부하였는데, 이진 세마포어와 카운팅 세마포어에 대한 학습을 주로 진행하였다. 운영체제의 자원 개수 조절과 관련된 요소들을 학습하며, 운영체제에 대해 더 깊게 이해할 수 있었던 시간이었다.</p> <p>- 손효림 이번 주차는 같은 주제로 저번 주에 이어서 학습하였다. Liveness 우선순위 제출 및 교착 상태 개념이 매우 흥미로웠다. 저번 주에 Semaphores가 잘 이해가 되지 않았는데 Mutex와 비교해 다시 짚어보면서 이해할 수 있어서 좋았다.</p> <p>- 이경화 이번 주차에는 지난주에 이어 동기화에 대한 심화 학습을 진행하였다. 임계구역에 대해 간단하게 복습을 진행한 후, Peterson 해결안에 대해 심화된 내용을 공부하였다. 그리고 Semaphore에 대해 더욱 구체적으로 알아보았다. 동기화라는 주제가 이제까지 배웠던 다양한 내용을 바탕으로하여 Peterson 해결안, Semaphore 등의 이해하기 까다로운 내용을 다루는 부분이라 처음 접했을 때는 소화하기에 버거운 느낌이 들었는데, 복습과 함께 심화 공부를 하면서 이해가 잘 안되었던 부분을 보완하면서 더 구체적으로 알 수 있게 된 것 같아 보람찬 스터디 시간이었다.</p> 
다음 학습계획	일정 : 11월 22일 12:00~14:45 (50주년기념관 516호)