

2023년 2학기 스터디그룹 주간학습보고서

그룹 명	MaveOS		
날 짜	11월 29일	시 간	12시 00분 ~ 14시 45분
스터디장소	50주년 기념관 516호	회차	_11_ 회차 모임
수강과목	운영체제	담당교수	강지훈
스터디	리더: 한아림		
참석자	그룹원: 손효림, 이경화		

학습문제

<메인 메모리>

- 배경
- 연속 메모리 할당
- 페이징
- 페이지 테이블의 구조
- 스와핑
- Intel 32비트와 64비트 구조

학습문제 해결과정

<메인 메모리>

#1. 페이징

1) 페이징이란?

- 페이징(paging) : 프로세스의 물리적 주소 공간이 연속되지 않는 것을 허용하는 메모리 관리 체계

2) 프레임 vs 페이지

- 프레임
 - 물리적 메모리를 나누는 고정 크기 블록
 - 프레임의 시작 주소와 오프셋이 결합하여 물리 메모리 주소가 됨
 - 오프셋은 참조되는 프레임 안에서의 위치
 - 프레임 크기는 하드웨어가 결정

- 페이지

- 논리 메모리를 나누는 동일 크기 블록
- 페이지 크기는 하드웨어가 결정
- 페이지 크기는 2의 거듭제곱이며 일반적으로 4KB ~ 1GB

3) 페이징 테이블의 구조

- CPU에서 생성된 모든 주소는 페이지 번호와 페이지 오프셋으로 나뉜다.
- 페이지 번호는 페이지 테이블에 대한 인덱스
 - 페이지 테이블은 물리적 메모리에 있는 각 프레임의 시작 주소를 포함

4) 페이징의 내부 단편화

- 메모리 요구량이 항상 프레임 크기의 배수일 수 없기 때문에 마지막 프레임은 대부분 내부 단편화가 발생한다.
 - 모든 프레임은 프로세스에게 할당될 수 있으므로 외부 단편화는 발생하지 않는다.

5) 페이징과 프로세스의 관계

- 프로세스가 로드될 때 페이지 테이블 생성
- 프로세스는 정보가 없는 다른 프로세스의 프레임에 접근 불가 (페이지 테이블을 통해서만 메모리에 접근하기 때문)
- OS는 모든 물리적 프레임 할당을 추적 (이를 위해 프레임 테이블 구조체를 관리)

6) 페이징에 대한 프로그래머의 인식과 실제

- 프로그래머의 인식
 - 메모리는 하나의 연속적인 공간
 - 메모리에는 자신이 실행/개발하는 프로그램만 있다고 생각
- 실제
 - 프로그램이 메모리의 여러 곳에 프레임 단위로 분산
 - 다른 많은 프로그램이 같이 로드된 상태
- 이 차이는 주소 변환 기술에 의해 해결

7) 하드웨어 지원 : 페이지 테이블의 하드웨어 구현

- 하드웨어 레지스터 세트
 - 가장 간단함
 - 각각의 레지스터가 컨텍스트 스위칭 중에 교체되어야 하므로 컨텍스트 스위칭 시간이 증가
 - 페이지 테이블이 작은 경우 적합하나 현대에선 부적합
- 페이지 테이블 기준 레지스터(PTBR, page-table base register)
 - 다른 페이지 테이블을 사용할 때 레지스터 값만 바꾸면 되므로 컨텍스트 스위칭 시간이 단축

8) Translation Look-aside Buffer(변환 색인 버퍼, TLB)


- 일종의 페이지 테이블의 캐싱 기술
- 각 항목은 key와 value 두 쌍으로 이뤄짐
- 페이지 테이블과 함께 사용됨
- 페이지 테이블 항목을 고르고 교체하는 방식은 다양함
- 특정 페이지 테이블 항목을 제거하지 않고 항상 유지하도록 지원하는 경우도 있음

9) 메모리 보호 : 각 프레임과 관련된 보호 비트

- 유효/무효 비트를 표현하는 비트를 하나 추가해 읽기-쓰기, 읽기 전용임을 각각 정의

10) 공유 페이지 : 공통 코드가 재진입 코드(reentrant code)인 경우 공유

- 공유 라이브러리로 둘 이상의 프로세스가 동시에 동일한 코드 실행 가능

학습성찰	
학습내용 이해도	100 %
<p>학습활동 돌아보기</p> <p>(좋았던 점, 아쉬운 점 활동모습 사진 추가)</p>	<p>- 한아림 이번 주차에서는 먼저 페이징에 대한 내용을 학습하였다. 본 내용의 학습을 통해 프레임과 페이지의 정의와 차이에 대해 면밀하게 파악할 수 있었다. 이 뿐만 아니라 CPU에서 생성된 주소가 어떻게 페이지 번호와 오프셋으로 나뉘어 들어가는지, 테이블 인덱스 번호는 어떻게 부여되는지에 대해서도 알 수 있었다. 그후 단편화 기능과 프로그래머의 작업 영역에 대한 학습도 진행하였는데, 이를 통해 CS와 하드웨어 지원 체계에 대한 이해도를 크게 높일 수 있었던 유익한 시간이었다.</p> <p>- 손효림 이번 주에는 페이징에 대해 공부하였다. 물리적 메모리 블록인 프레임과 비교하는 것으로 기본 방식과 특징을 탐구하고 내부 단편화 문제부터 프로세스와의 관계, 메모리에 대한 프로그래머의 인식과 실제의 차이, 이에 대한 해결책, 하드웨어 지원과 변환 색인 버퍼와 메모리 보호, 공유 페이지에 대한 내용으로 교안을 제작하였다. 이번 주차 수업 내용이 방대해 요약하는데 조금 힘들었던 것 같다. 마지막 주차를 맞아 후련하고 아쉬운 기분이다.</p> <p>- 이경화 이번 주차 스터디에서는 메인 메모리라는 큰 주제 안에서 페이징이라는 세부 주제를 중심으로 학습했다. 물리적 메모리를 나누는 구조인 프레임과 논리 메모리를 나누는 구조인 페이징을 비교하는 과정에서 페이징에 대한 이해를 더 명확하게 할 수 있어서 좋았다. 이해를 돕기 위한 그림이 많이 첨부되었는데 해당 그림을 외우다시피 반복학습하면 더 깊이 학습할 수 있을 것 같다. 이렇게 스터디 마지막 주차를 마무리 하였는데 혼자 공부하는 것보다 학습 스케줄을 더 잘 관리할 수 있었던 점이 가장 좋았고, 함께 공부하는 스터디원들이 있다는 점이 어려운 내용을 이해하려고 노력하는 데에 큰 힘이 되었다.</p> 
다음 학습계획	없음