

2023년 2학기 스터디그룹 주간학습보고서

그룹 명	MaveOS		
날 짜	11월 22 일	시 간	12시 00분 ~ 14시 45분
스터디장소	50주년 기념관 516호	회차	10 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. HW와 메모리 배경

1) 기본 하드웨어

- 프로세스가 접근 가능한 합법적 메모리 주소의 범위를 결정하고 프로세스가 합법적인 주소에만 접근할 수 있도록 하는 기능이 필요
- 기본 하드웨어 : 기준 레지스터(가장 작은 합법적인 물리 메모리 주소의 값), 상한 레지스터(영역의 크기 저장)
- 프로세스는 기준 레지스터의 값에서 상한 레지스터의 값을 더한 값까지의 모든 주소를 접근할 수 있음

2) 기준 및 상한 레지스터를 사용해 메모리를 보호하는 간단한 예

- OS는 하나의 연속적인 기본 메모리 범위를 프로세스에게 할당
- 기준 레지스터에는 프로세스의 가장 낮은 주소가 저장
- 상한 레지스터에는 할당된 바이트 수가 포함됨
- 기준/상한 레지스터를 통해 사용자 모드에서 생성되는 모든 주소 확인

3) 메모리 공간의 보호

- 이러한 시스템은 사용자 프로그램이 운영체제나 다른 사용자의 코드 및 데이터 구조를 수정하는 것을 방지함
- CPU가 사용자 모드에서 생성된 모든 주소를 레지스터와 비교
- 사용자 모드에서 실행되는 프로세스가 OS 메모리 또는 다른 프로세스의 메모리에 접근하려고 하면 trap이 발생하며, 이를 치명적인 오류로 간주

4) 주소 바인딩

- 프로그램은 이진 실행 파일 형태로 디스크에 저장
- CPU는 PC(Program Counter)가 지시하는 대로 메모리에서 다음 수행할 명령어를 가져와서 실행함
- 프로그램이 실행되어 메모리에 로드될 때, 메모리의 어느 영역을 사용할지 주소를 결정해야 함

5) 프로그램 실행 시

- 디스크에 저장되어 있던 프로그램을 메모리로 가져와 프로세스 형태로 배치해야 함
- 프로세스가 실행되면 메모리에서 명령 및 데이터에 접근함
- 프로세스의 마지막 명령어가 실행되고 종료됨
- 종료된 프로세스가 사용하던 메모리 영역은 다른 프로세스가 사용할 수 있도록 회수됨
- 대부분의 시스템은 사용자 프로세스가 메모리 내 어느 부분으로도 배치될 수 있도록 지원함

6) 특징

- 컴퓨터가 0번지부터 주소 공간을 사용한다고 하더라도 사용자 프로세스의 시작 주소가 0번지일 필요는 없음
- Source program(Source file)에서 주소는 숫자가 아닌 변수와 같은 심볼 형태로 표현됨
- 컴파일러는 이 심볼 주소를 재배치 가능한 상대 주소로 바인딩시킴
- 링커 또는 로더가 재배치 가능한 주소를 절대 주소로 바인딩함
- 각 주소 바인딩 작업은 한 주소 공간을 다른 주소 공간으로 매핑하는 작업을 뜻함

#2. 메인 메모리와 바인딩 기법

1) compile time 바인딩

- 물리적 메모리 주소가 컴파일 단계에서 결정됨
- 프로세스가 메모리 내에 들어갈 위치를 컴파일 시 미리 알 수 있으면 컴파일러는 절대 코드(Absolute code)를 생성할 수 있음
- 컴파일 단계 대 결정된 주소가 결국 메모리에 올라가는 주소가 됨
- 위치가 변경되어야 한다면 다시 컴파일 되어야 함
- 다수의 프로세스의 실행과 종료가 반복되는 변칙적인 환경에서 사용하는데 제한적임
- 아두이노 > 메모리에 1개의 프로세스만 허용되는 환경

2) load time 바인딩

- 프로그램이 실행될 때 물리적 메모리 주소가 결정됨
- 컴파일러는 이진 코드를 재배치 가능 코드(relocatable code)로 만들어야 함
- 심볼과 진짜 주소와의 바인딩은 프로그램이 메인 메모리로 실제로 적재되는 시점에 이루어짐
- 재배치 가능 코드는 시작 주소가 변경될 때 사용자 코드를 다시 적재해야 함
- OS에 의해 메모리가 변칙적으로 바뀌는 상황에 대응하지 못함

3) execution time 바인딩

- 프로세스가 실행 중에 메모리 내의 한 세그먼트로부터 다른 세그먼트로 옮겨질 수 있음
- 실행 중 주소 바인딩이 수행되기 때문에 OS에 의해 메모리가 바뀌어도 탄력적으로 수용 가능함
- 프로세스 실행 중에 주소가 바인딩되어야 하기 때문에 특별한 하드웨어가 필요함

#3. 메인 메모리와 주소 공간

1) 논리 & 물리 공간

- CPU가 프로세스를 실행할 때 생성하고 참조하는 주소는 일반적으로 논리 주소(Logical Address) 또는 가상 주소(Virtual Address)라고 함
- 메모리에서 취급되는 주소는 일반적으로 하드웨어 주소라고도 하는 물리 주소(Physical Address)라고 함
- Compile 또는 Load time 주소 바인딩을 하게 되면, 논리 주소와 물리 주소가 같음
- 메모리 주소 레지스터(MAR-Memory Address Register)에 로드된 주소
- Execution time 주소 바인딩을 하게 되면 논리 및 물리 주소는 서로 다름
- 프로그램에 의해 생성된 모든 논리 주소 집합을 논리 주소 공간(Logical Address Space)라고 함
- 논리 주소 공간과 매핑되는 모든 물리 주소 집합을 물리 주소 공간(Physical Address Space)라고 함
- 프로그램의 실행 중에는 논리 주소를 물리 주소로 바꿔줘야하는데 이러한 매핑 작업은 하드웨어 장치인 MMU(Memory Management Unit)가 수행

#4. 메인 메모리와 메모리 할당

1) 연속 메모리 할당

- 메모리는 일반적으로 운영체제를 위한 영역 & 사용자 프로세스를 위한 영역으로 구성
- 컴퓨터 시스템에서 운영체제는 낮은 메모리 주소나 높은 메모리 주소에 배치될 수 있음
- 메모리에 적재되길 기다리는 프로세스에 사용 가능한 메모리를 할당하는 방법을 고려해야 함
- 또한, 각 프로세스 메모리 영역이 침범되지 않도록 메모리 보호 문제를 해결해야 함
- 일반적으로 많은 운영체제들은 높은 메모리 주소에 배치됨
- 만일 시스템이 상한 레지스터와 재배치 레지스터를 가지고 있다면 프로세스가 소유하지 않은 메모리에 접근할 수 없도록 할 수 있음
- 상한 레지스터 : 논리 주소의 범위
- 재배치 레지스터 : 가장 작은 물리적 주소의 값

학습성찰	
학습내용 이해도	99 %
<p>학습활동 돌아보기</p> <p>(좋았던 점, 아쉬운 점 활동모습 사진 추가)</p>	<p>- 한아림 이번 스터디에서는 메모리의 주요 요소인 '메인' 메모리에 대한 학습을 진행하였다. 메인 메모리가 필요하게 된 배경을 먼저 학습하였고, 그 후 연속 메모리 할당과 페이징, 페이징 테이블 구조와 스와핑에 대한 내용까지 공부하였다. 이번 주차 스터디 내용은 공부하면 할 수록 컴퓨터 요소들은 모두 '보안' 강화를 중점으로 발전해왔다는 것을 느낄 수 있었다. 특히 주소 바인딩 부분과 관련하여 특권 명령과 각 모드 간 권한 관리, 메모리 영역의 보호 등의 궁극적인 목표들이 현재 컴퓨터 과학 구조를 완성시켰다는 것이 참 신기하게 다가왔다. 이번 스터디는 메모리의 구조가 왜 현대처럼 나누어지게 되었는지, 왜 메모리와 Computer 요소들이 지금처럼 작동하는지에 대한 이해도를 높일 수 있었던 좋은 시간이었던 것 같다.</p> <p>- 손효림 이번 주차 스터디에서는 메인 메모리를 대주제로 하여 바인딩, 주소 공간, 메모리 할당과 연속성에 대해 학습하였다. 작년 컴퓨터 구조 때 메모리 지역성에 관한 부분을 재미있게 학습했던 기억이 있는데 이번 스터디에서는 바인딩에 대한 부분이 흥미로워 내가 메모리에 관심이 있다고 생각하게 되었다. 최근 노트북을 사려는 친구들에게 컴퓨터 사양에 대한 조언을 해주고 있는데 메인 메모리에 대해 학습한 부분이 도움이 되길 바란다.</p> <p>- 이경화 이번 스터디의 주제는 '메인 메모리'였다. 이전 스터디에서 CPU 스케줄링과 동기화에 대해 많은 학습량을 다룬 뒤, 실제로 메모리에서 코드와 데이터 등을 어떻게 관리하는지 학습하니 자연스럽게 이해할 수 있었다. 이번 학습 내용에서는 그림이 많이 등장했는데, 복습 차원에서 그림을 직접 그려보며 공부한 내용을 다시 확인해보면 학습 효과를 높일 수 있을 것 같다.</p> 
다음 학습계획	일정 : 11월 29일 12:00~14:45 (50주년기념관 516호)