**Pintos Project 4: Virtual Memory**

담당 교수 : 김영재

이름 / 학번 : 정은교 20211590

개발 기간 : 2023.12.1~2023.12.10

1. **개발 목표**

* 해당 프로젝트에서 구현할 내용을 간략히 서술

page fault handling을 구현하는 것이 목표인데 이는 virtual memory 구현 즉 vm directory의 page.c, page.h, swap.c, swap.h, frame.c, frame.h 파일을 생성하여 page table, swapping, stack growth를 구현한다.

1. **개발 범위 및 내용** 
   1. **개발 범위**

* 아래 각 항목 개발의 필요성 또는 개발 시 기대되는 결과를 간략히 서술
  1. Page Table & Page Fault Handler

Page fault가 발생하였을 때 invalid access가 아니라 swapping을 구현하였다면, disk에 없다 하여도 disk에 해당 page가 있는 지 없는지 알 수 있기 때문에 disk에 해당 페이지가 있다면 로드하는 방식의 결과를 얻을 수 있다.

* 1. Disk Swap

Memoery 요청을 받은 상황에서 해당 process가 필요한 physical memory가 부족할 때, disk로의 swap-out이 일어나고, 이때 선택 당할 page는 second chance로 구현되어, disk swap을 할 수 있다.

* 1. Stack Growth

기존의 stack 크기인 4KB를 넘어선 접근이 발생하여 stack에서의 page fault가 일어났다면, 또 다른 stack page를 load하여 stack growth 처리를 한다.

* 1. **개발 내용**
* 아래 항목의 내용만 서술
  1. Page fault가 발생하는 이유와 이를 handling하는 전반적인 과정을 서술

가상 주소에 해당하는 물리 주소가 존재하지 않으면 페이지 폴트가 발생한다. 이는 페이지 테이블을 통해 해당 가상 주소에 대응되는 프레임을 찾지 못했을 때 처리되며, 디스크로부터 해당 프레임에 대응하는 페이지를 가져와 메모리에 적재한다. 이때 주소가 세그먼트 영역에 속하는지 등의 유효성 검사를 수행하고, 페이지 테이블을 업데이트하여 가상 주소와 물리 주소를 매핑한다.

* 1. Disk swap 발생 시 사용한 page replacement algorithm에 대해 서술

페이지 폴트가 발생했을 때, 메모리가 가득 찬 경우가 있을 수 있다. 이때는 기존에 메모리에 있던 프레임을 디스크로 대체하여 공간을 확보하고자 하는 페이지를 메모리에 적재한다. 이를 위해 swap 과정이 필요하며, LRU(Least Recently Used) 알고리즘을 사용하여 가장 최근에 사용되지 않은 프레임을 선정하여 디스크로 대체한다.

* 1. Stack growth 구현 시 stack 확장 여부를 판단할 수 있는 방법에 대해 서술

페이지 폴트가 발생하여 스택을 확장하려 할 때, 접근이 유효한지 판별해야 한다. 이는 사용자 메모리의 스택 영역이거나 스택 프레임에 있는 주소인지 확인해야 한다. 스택 포인터(esp)를 사용하여 이를 확인하며, 스택 영역을 벗어난 경우 세그멘테이션 폴트가 발생하므로 이에 대한 처리를 수행한다.

1. **추진 일정 및 개발 방법**
   1. **추진 일정**

* II. A. 개발 범위를 포함하여 구현 내용에 대한 일정 작성
* 12/1~ 12/15 : Page Table & Page Fault Handler 구현 & system call 예외 처리
* 12/6~12/9 : disk swap & stack growth 구현
* 12/10: 보고서 작성
  1. **개발 방법**
* II. B.의 개발 내용을 구현하기 위해 각각에 대해 다음 사항들을 포함하여 설명
  + 수정해야하는 소스코드

Vm/frame.c, frame.h, page.c, page.h, swap.c, swap.h

Makefile.build & devices/timer.c & threads/init.c, interrupt.c, thread.c, thread.h

Userprog/exception.c, pagedir.c, process.c, syscall.c, syscall.h

* + 수정하거나 추가해야 하는 자료구조

Struct thread, struct vm\_entry

* + 수정하거나 추가해야 하는 함수

Vm direct 하의 함수들

Process.c: grow\_stack, mm\_handle, verify\_stack 등

1. **연구 결과**
   1. **Flow Chart**

* II. B. 개발 내용의 각 항목에 대하여 Flow Chart 작성

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

* 1. **제작 내용**
* II. B. 개발 내용의 각 항목에 대하여 실질적으로 구현한 코드의 관점에서 작성 (구현 내용, 알고리즘 등을 명확히 서술할 것)
  + 구현에 있어 Pintos에 내장된 라이브러리나 자체 제작한 함수를 사용한 경우 이에 대해서도 설명
* 개발 중 발생한 문제나 이슈가 있으면 이를 간략히 설명하고 해결한 방식에 대해 설명

Threads/thread.h vm 추가

**폰트, 그래픽, 스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명**

vm/page.h 자료 구조 추가

**텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명**

vm/page.c

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Vm/swap.h

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Vm/swap.c

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Vm/frame.h

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Vm/frame.c

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Userprog/excepetion.c 위에서 선언한 함수들(verify\_stack, grow\_stack, handle\_mm\_fault) 함수를 사용해 예외처리 추가한다.

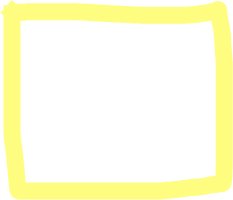
텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Userprog/process.c

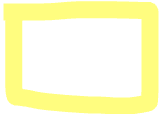
텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명



텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명



텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명



* 1. **시험 및 평가 내용**
* (채점 대상 테스트 케이스에 해당하는) make check 수행 결과를 캡처하여 첨부

텍스트, 스크린샷, 폰트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명