**Pintos Project 4: Virtual Memory**

담당 교수 : 김영재

이름 / 학번 : 임나현 / 20211582

개발 기간 : 2024.12.01 – 2024.12.22

1. **개발 목표**

* 해당 프로젝트에서 구현할 내용을 간략히 서술

이전까지의 pintos 과제에서는 main memory size의 크기만큼만 프로그램을 작성할 수 있었으나 이번 project4에서는 virtual memory paging을 구현하여 disk와 memory 상의 swapping을 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 fault handler에서는 memory access를 하려는 포인터가 main memory 위치를 가리키지 않는 경우, disk에 존재하는 virtual address인지 확인해야 한다. 해당 address가 가리키는 page가 valid한지 확인하기 위해서 page table entry를 확인하고 pte가 있다면 disk에 있는 내용을 memory로 swaping해야 한다. 만약, pte가 없다면 segmentation fault 처리를 한다.

* 1. Disk Swap

main memory에 공간이 없는 경우, 사용 빈도가 낮은 data를 disk로 저장하기 위해 swap out을 구현한다. 현재 disk에 있는 data를 접근하고 싶은 경우 다시 main memory에 불러오기 위해 swap in 동작 또한 구현해야 한다.

* 1. Stack Growth

초기값으로 stack을 위한 공간을 많이 할당하지 않으나, Page fault를 handle할 때 stack growth를 통해 stack의 크기를 증가시켜 frame을 할당할 수 있도록 구현해야 한다.

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

Memory 접근이 이루어질 때 page가 main memory에 위치해 있지 않는 경우 page fault가 발생한다. 이를 handle 하기 위해서는 disk 상에 있는 data를 memory 상으로 swap하고 해당 페이지를 제대로 접근할 수 있도록 해야 한다. 이를 위해 page table entry가 필요하다.

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

여러 가지 알고리즘 중에서 second chance algorithm을 사용한다. accessed bit을 통해 0이면 해당 page를 교체하고, 1인 경우 accessed bit를 0으로 바꿔 기회를 두 번 주는 방식이다. 교체할 page가 결정될 때까지 계속 page를 찾기 위해 다음 page로 넘어간다.

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

page fault가 일어났을 때 접근하려고 하는 주소가 stack 확장이 가능한 유효 범위 내에 있는지 판단하고 stack growth 함수를 통해 stack을 확장한다. 최대 크기는 8MB까지이고 유효한 접근이 아닐 경우 예외 처리한다.

1. **추진 일정 및 개발 방법**
   1. **추진 일정**

* II. A. 개발 범위를 포함하여 구현 내용에 대한 일정 작성

2024.12.1 - 2024.12.10: manual 읽고 disk swapping에 대한 이해

2024.12.10 – 2024.12.19: page, frame, swap 파일 구현

2024.12.19 – 2024.12.21: exception.c, stack growth 이해

2024.12.22: 보고서 작성

* 1. **개발 방법**
* II. B.의 개발 내용을 구현하기 위해 각각에 대해 다음 사항들을 포함하여 설명
  + 수정해야하는 소스코드
  + 수정하거나 추가해야 하는 자료구조
  + 수정하거나 추가해야 하는 함수
  1. Page fault

vm 폴더에 새롭게 page.c와 page.h 파일을 추가하여 page들을 관리할 수 있는 page table entry 구조체를 선언한다. 구조체 안에는 read byte, address, swap의 유무 등을 위한 변수가 필요하다. hash list를 활용하면 빠르고 효율적으로 탐색 및 접근을 할 수 있다. hash list를 초기화하고, entry를 삽입, 삭제할 수 있는 함수를 page.c에 구현한다. process.c에서는 disk에 있는 page를 main memory에 mapping할 수 있는 함수가 필요하다. exception.c에서는 page fault가 발생했을 때 page\_fault 함수에서 page fault를 handling할 수 있도록 구현해야 한다.

* 1. Disk swap

vm 폴더에 새롭게 frame.c와 frame.h 파일을 추가한다. frame 관리를 위해서는 lock을 이용하여 다른 함수와의 충돌을 방지하기 위한 보호가 필요하다. frame table entry를 삽입하거나 삭제하는 함수가 필요하며, disk swap 알고리즘으로 사용할 second chance algorithm의 구현 또한 필요하다. Swap을 영역을 먼저 초기화하는 함수의 구현이 필요하고 swap으로부터 읽거나 쓰는 함수가 필요하다.

* 1. Stack growth

Page fault를 handle할 때 유효한 접근에 한해서 stack growth를 하는 함수를 구현하여 처리한다. exception.c에서 page fault handling을 처리할 때 stack growth가 필요한 경우 해당 함수를 호출한다. Virtual address를 커널 주소로 바꾼 mapping address 또한 page table에 저장이 되어야 한다.

1. **연구 결과**
   1. **Flow Chart**

* II. B. 개발 내용의 각 항목에 대하여 Flow Chart 작성
  + 1. Page fault

텍스트, 친필, 폰트, 스케치이(가) 표시된 사진

자동 생성된 설명

* + 1. Disk swap

텍스트, 그림, 스케치, 친필이(가) 표시된 사진

자동 생성된 설명

* + 1. Stack growth

친필, 그림, 스케치, 라인 아트이(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

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

새로 추가한 파일에 대해서 makefile.build에 make file될 수 있도록 추가한다.

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

Page table entry 구조체의 형태이다. 관리를 위한 read\_bytes, mem\_load, read\_only 등의 변수가 선언이 되어 있다.

**텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명**

Hash\_entry를 통해 주어진 hash\_elem에서 pagetable 항목을 가져와 특정 메모리를 제거하는 함수이다 lock을 통해 frame table에 대해 잠금한다. Read\_success 함수는 파일에서 읽기가 성공했는지 여부를 나타내는 함수이고 page\_load 함수를 통해 특정 주로 데이터를 load하는 함수이다.

**텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

process.c에서 frame을 allocate하는 함수이다. palloc\_get\_page 함수를 호출하여 페이지를 할당하고 실패하는 경우 예외 처리한다. Dirty 상태이거나 page access이 이루어지지 않는 경우 예외 처리한다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Disk swap에서 data를 memory에서 disk로 옮기는 swap out함수, disk에서 memory로 다시 load하는 swap in 함수를 swap.c에 구현하였다. 초기에 swap을 init하는 함수 또한 있으며 lock을 사용하여 disk swap에서 충돌을 방지하도록 한다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Process.c함수에서 setup\_stack 시 supplemental pagetable값도 초기화하고 initial 값을 할당할 수 있도록 구현한다.

**텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

Exception.c에서 page fault를 handling하고 유효한 주소 값 범위에 있으나 stack의 크기가 부족할 경우 grow\_stack함수를 호출하여 stack를 확장한다.

* 1. **시험 및 평가 내용**

1. (채점 대상 테스트 케이스에 해당하는) make check 수행 결과를 캡처하여 첨부

텍스트, 스크린샷, 폰트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

총 16개의 test case 중에서 12개의 test case에 성공했다.