**Pintos Project 4: Virtual Memory**

담당 교수 : 김영재

이름 / 학번 :20171628 김현우

개발 기간 :11/10~11/27

1. **개발 목표**

이전 프로젝트에서는 page fault가 일어나면 종료가 되었지만 이번 프로젝트에서는 page fault에 대한 handling을 추가하여 조금 더 안정적이게 실행되도록 핀토스를 수정한다. 또한 memory size에 또한 process가 제한되는 것을 virtual memory를 구현하여 제한을 없앤다.

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

프로세서가 page\_fault 를 실행하면 kernel 은 page\_fault handler를 불러서 page\_fault를 handling 한다. 만약에 유효한 주소라면 frame을 받아서 page를 저장한다. frame으로 부터 데이터를 가져온다. 실제 물리적 주소에 가상 주소를 할당해준다.

만약 유효하지 않다면 프로세스를 종료하고 모든 자원들을 free해준다.

이렇게 하면 page\_fault인 상황이 오더라도 프로그램이 그냥 종료 되지 않고 제대로 실행이 될 수 있다.

* 1. Disk Swap

Swap을 구현한다. process에 할당해 줄 physical memory가 부족할 때 disk로 swap out이 일어난다. 이때 page replacement algorithm을 활용해서 swap 해줄 page를 결정한다. Swap table 을 이용해서 현재 사용하는 slot 과 아닌 slot 을 관리한다. 이렇게 하면 disk swap 이 일어나서 page replacement가 일어난다.

* 1. Stack Growth

이전 프로젝트는 program 의 크기가 제한이 되었으나, 이번 프로젝트를 통해 stack 보다 크기가 크다면 추가로 page를 할당할 수 있게된다. 단 page fault 중에서 stack 에 있는 page fault 같은 경우는 다른 stack page를 할당한다.(확장도 최대 8MB 까지 )

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

Page fault는 찾으려는 페이지가 메모리에 존재하지 않아서 생기는 문제이다. Page\_fault가 발생했을 시 가르키는 주소가 유효한 주소라면 frame을 받아서 page를 저장한고 frame으로 부터 데이터를 가져온다. 그리고 실제 물리적 주소에 가상 주소를 할당해준다. 만약에 공간이 부족할 시 replacement algorithm을 활용해서 물리페이지에 할당한다.

만약 유효하지 않다면 growable region인 여부를 확인하고 만약에 아니라면 프로세스를 종료하고 모든 자원들을 free해준다. 만약에 growable region 일 시는 user stack을 확장하고 process 를 재개한다

* 1. Disk swap 발생 시 사용한 page replacement algorithm에 대해 서술
  2. Stack growth 구현 시 stack 확장 여부를 판단할 수 있는 방법에 대해 서술

Page\_fault 상황일 때 stack acess 에 해당하는 주소인 경우에만 stack growth라고 가정한다. 매뉴얼에 나오듯이 push 계산인 경우의4바이트이거나 pushA 연산일때 32바이트 아래로 내려가는 경우일 때 stack growth 라고 판단한다.

1. **추진 일정 및 개발 방법**
   1. **추진 일정**

* II. A. 개발 범위를 포함하여 구현 내용에 대한 일정 작성

11/20 ~ 11/25 stack grow 구현

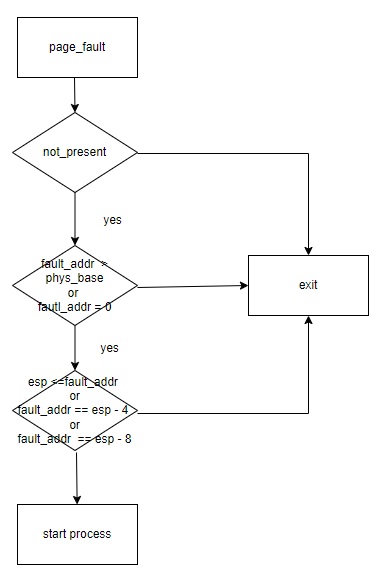
12/26 보고서 작성

* 1. **개발 방법**
* II. B.의 개발 내용을 구현하기 위해 각각에 대해 다음 사항들을 포함하여 설명
  + 수정해야하는 소스코드
  1. Makefile.build에 65번째줄에 새로 컴파일 하는 파일들을 넣는다.
  2. Page\_fault 함수를 수정하여서 바로 프로그램을 종료하는 대신에 page fault handling을 통해서 stack grow 로 간주되는 상황일 때 stack grow,를 한다.

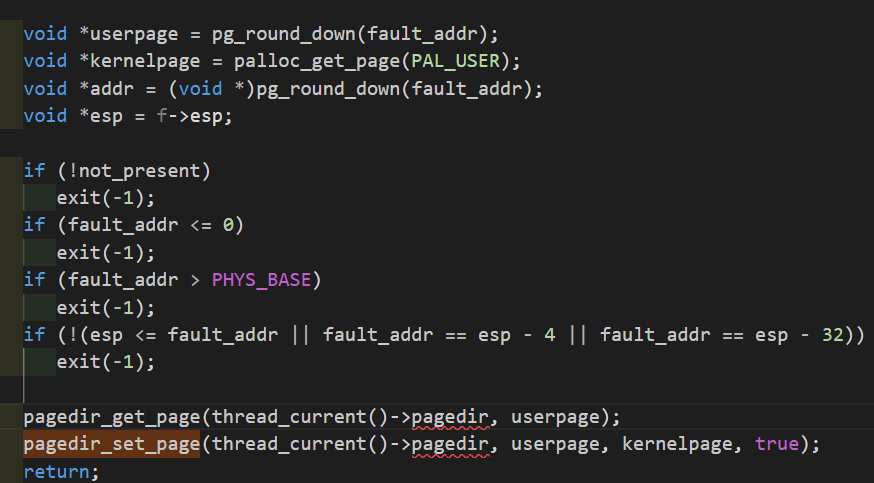
1. **연구 결과**
   1. **Flow Chart**

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

Stack growth

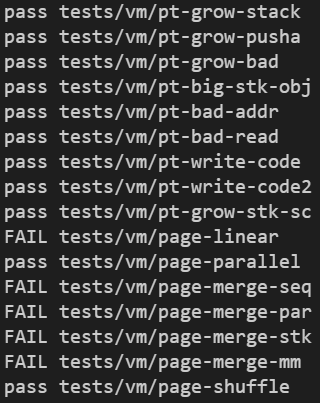


* 1. **제작 내용**
* II. B. 개발 내용의 각 항목에 대하여 실질적으로 구현한 코드의 관점에서 작성 (구현 내용, 알고리즘 등을 명확히 서술할 것)



위에서 설명한 대로 해당 page가 memory에 not\_present하면 exit(-1)을 소환해서 프로세스를 나가고 fault address가 0초과 phsy\_base 이하의 주소에 있는지 확인하고 esp 의 주소를 확인해서 만약에 stack\_growth라고 판별이 되면 stack을 늘려준다.

Page\_get\_page와 page\_set\_page를 통해 가상 주소인 page를 통해서 실제 phsyscal 주소를 찾는다.

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

Stack growth 에 관해서는 통과가 잘되는 모습이다.