운영체제 4 차과제

(1) 과제 설명

- **목표:** Copy-on-Write (CoW) 기능 구현 및 실험

- 과제를 위한 배경 지식:
    - x86 아키텍쳐의 CR3 레지스터는 top-level 페이지 디렉토리를 가리키고 있음. 그리고 페이지 테이블의 entry 들은 TLB 라고 불리는 하드웨어에 cache 되어 있음. 따라서, 페이지 테이블의 어떠한 변화가 생기면, 그동안 cache 되어있던 페이지 테이블의 변화가 생겼다는 것을 알려줘야 함. 이를 위해 아래와 같이 xv6 에서 제공하는 lcr3 함수를 호출해야 함.
        - lcr3(V2P(p->pgdir));
        - 위 함수는 switchuvm 함수에서도 사용되며, context switching 시 매번 호출됨
- 1 단계: getNumFreePages 시스템 콜 추가
    - kalloc.c 파일에 uint num_free_pages; 전역변수 선언
    - kinit1 함수에서 0 으로 초기화
    - kalloc 함수에서 --, kfree 함수에서 ++
- 2 단계: 각 메모리페이지 들을 대상으로 reference counter 적용 (물리메모리 주소 사용)
    - kalloc.c 파일에 uint pgrefcount[PHYSTOP >> PTXSHIFT]; 전역변수 선언
    - get_refcount, inc_refcount, dec_refcount 함수 구현
    - freerange 함수에서 pgrefcount 배열 0 으로 초기화
    - kfree 함수에서 dec_refcount 함수 호출
    - copyuvm 함수에서 inc_refcount 함수 호출
    - kalloc 함수에서 refcount 1 로 설정
- 3 단계: copyuvm 수정
    - fork 함수에서 copyuvm 함수를 호출함

- copyuvm 함수를 호출해서 새로운 프로세스가 생성될 때 메모리 페이지들을 생성하고, 실제로 복사하는 것이 아니라 부모 프로세스의 메모리 페이지들을 사용하도록 변경하고 Write 권한을 없애야 함
- 그리고 inc_refcount 함수 호출하여 해당 메모리 페이지들의 reference counter 증가시킴

- 4 단계: page fault handler 구현
  - trap.c 파일의 trap 함수 수정하여 page fault 발생시 핸들러 호출하도록 수정
  - vm.c 파일에 pagefault() 함수 구현
  - pagefault 함수는 아래 기능들 수행
    - rcr2() 함수를 호출하여 page fault 가 발생한 VA 읽어 들인다.
    - 해당 가상 주소의 page table entry 확인후 PA 찾는다.
    - PA 를 이용하여 해당 메모리 페이지의 reference counter 값 확인
    - reference counter 가 1 보다 큰 경우 새로운 페이지를 할당 받아서 복사
      - 새로운 페이지를 할당 받았기 때문에, refcount 값 감소시켜야함
    - reference counter 가 1 인 경우 현재 pte 의 write 권한만 추가
    - 마지막으로 lcr3(V2P(curproc->pgdir)); 호출

(2) 제출기한

2024 년 12 월 6 일 자정

(3) 제출방법

스마트캠퍼스를 통해서 제출

(4) 제출물

- 아래 파일들을 하나의 압축 파일로 제출 해야함.
- 압축파일의 이름은 반드시 "os4_학번.zip" 으로 해야함
  - 보고서 (반드시 자신의 글로 작성할 것)
    - 소속, 학번, 이름
    - 개발환경
    - 수정 및 작성한 소스코드에 대한 설명
    - 과제 수행 중 발생한 문제점과 해결 방법
  - 직접 수정하고 작성한 소스파일 전체 (XV6 소스코드 전체 올리지 말 것)

- 직접 수정하고 작성한 파일들에 대한 간략한 설명을 "ReadMe.txt"의 파일이름의 텍스트 파일로 작성