OS_요구 페이징

Demand Paging

- 프로세스 이미지는 backing store에 저장
- 프로세스는 페이지의 집합
- 지금 필요한 페이지만 메모리에 올린다(load) => 요구되는(demand) 페이지만 메모리에 올린다

하드웨어 지원

- valid 비트 추가된 페이지 테이블
- backing store(= swap device)
- 가상 메모리를 사용하면 메모리 크기가 제한적이라도 최대 크기는 하드디스크 크기만큼 보이게 할 수 있다.

페이지 결함 (= 페이지 부재) Page Fault

- 접근하려는 페이지가 메모리에 없다. (invalid)
- 처음에는 사용하지 않는 페이지라 메모리에 로드하지 않았는데 실행하다보니 필요해짐(ex: 오류처리) => 페이지 테이블에서 해당 페이지는 invalid (0 비트) 상태임
- Backing store에서 해당 페이지를 가져온다
- Steps in handling a page fault

동작과정

- CPU에서 주소 냈는데 valid 비트가 0이면 Interrupt 신호를 CPU로 보냄
- CPU가 Interrupt 받으면 OS의 Interrupt 처리 루틴으로 이동
- 하드디스크에서 필요한 페이지를 메모리에 로드
- Page Fault 발생한 page table의 page number를 방금 로드한 메모리 주소로 바꾸고 valid 비트를 1로 바꿈

Swapping과의 차이점

- Demand paging은 메모리에 page가 로드되고 해제 되는 것
- Swapping은 프로세스 자체가 메모리에 로드되고 해제 되는 것

유효 접근 시간

- 프로세스마다 valid 비트에 의한 읽기 속도 차이가 있을 것 (유효하지 않으면 하드에서 찾아 다시 올려야하니까)
- 평균 읽기 속도를 유효 접근 시간이라고 함
- p: 페이지 fault가 일어날 확률

- $T_{eff} = (1-p)T_m + pT_p$
  - T_{eff}: 유효 접근 시간
  - T_m: 메모리 읽는 시간 => 실제로는 페이지 테이블 읽는 시간도 포함이지만 작으므로 무시
  - T_p: page fault가 발생하여 소요되는 시간(하드디스크에서 필요한 페이지를 찾아 메모리에 로드하고 페이지 테이블 갱신하여 다시 읽는 시간, 대부분의 시간이 하드디스크에서 필요한 페이지를 찾는데 걸림)
- 예제
  - T_m = 200 nsec, T_p = 8msec (seek time + rotational delay + transfer time => 하드디스크)
  - p = 1/1000 => T_{eff}는 40배 느려짐
  - p = 1/399,990 => T_{eff}는 10% 느려짐

지역성의 원리

Locality of reference

- CPU가 참조하는 주소는 Locality => 몰려있다
- 메모리 접근은 시간적, 공간적 지역성을 가진다
  - 시간적 지역성: 근접 시간 대에서 읽은 메모리 주소는 비슷하다(반복문)
  - 공간적 지역성: 이전에 읽었던 메모리 주소와 인접한 주소를 읽는다
- 실제 페이지 부재 확률은 매우 낮다
- 그럼에도 페이지 부재시 시간이 너무 오래 걸림
- 다른 방법
  - HDD는 접근 시간이 너무 길다 => swap device로 부적합
  - SSD 또는 느린 저가 DRAM 사용