Page Sharing

이름	양희범
날짜	@2023년 9월 11일

Page Sharing

- 여러 프로세스가 특정 page 공유
- 공유 가능한 page
  - Procedure pages
    - Pure code(rentant code) : 수행되는 동안 절대 변하지 않는 코드
  - Data page
    - Read-only data
    - Read-write data : 병행성 제어 기법과 같이 사용

Page Protection Bit

- 여러 프로세스가 Page 공유할 때, Protection bit 사용
  - Vaild 타당 비트 - 메인 메모리에 적재 되어 있는지
  - Read 읽기 비트 - 읽기 여부
  - Write 쓰기 비트 - 쓰기 여부
  - Execution 실행 비트 - 실행 여부

Segementation System

- segement : 서로 다른 크기를 닺는 논리적인 단위
- 서브 루틴이나 함수 등의 단위로 나누거나 행렬이나 스택등의 대단위 자료 구조 등에 이름을 부여하여 적재
- 메모리를 미리 분할하지 않음

- 세그먼트 공유와 보호가 용이함
- 주소 mapping, 메모리 관리 → high overhead

Direct Mapping

1. 프로세스의 SMT가 저장되어 있는 주소 b에 접근
2. 해당 SMT에서 segment를 메모리로 적재 후 주소 확인
3. 찾아진 entry의 존재 비트 검사
    a. Residence bit = 0 - missing, swap device에서 해당 segment를 메모리로 적재 후 주소 확인
    b. d가 segement보다 큰 경우(d>1k), segment overflow exception 처리 모듈 호출
    c. 허가 되지 않은 연산일 경우(protection bit field 검사), segment protection exception 처리 모듈 호출
4. 찾은 주소와 가상 주소의 변위 d를 사용하여 실제 주소 r 확인
5. r로 접근

Paging	Segementation
고정된 크기의 block(page)로 분할	가변적인 논리 단위 segement로 분할
관리면에서 낮은 overhead	관리면에서 높은 overhead
내부 단편화 현상 발생	외부 단편화 현상 발생
공유와 보호가 복잡	공유와 보호의 지원이 편리함
필요한 페이지만 페이지 프레임에 적재하여 사용 → 메모리의 효율적 활용	필요한 세그먼트만 메모리에 적재하여 사용 → 메모리의 효율적 활용

Hybrid System

- 프로그램을 segment로 나눔
- segement를 page로 나눔
- page 단위로 적재
- V = (s, p, d) → (segment, page, displacement)
- SMT, PMT 모두 사용
- Table이 늘어나는 만큼 메모리 소모가 크다

- mapping 과정이 복작함 → 접근 시간도 늘어남
- 외부 단편화 X, 내부 단편화 O