1. 가상메모리

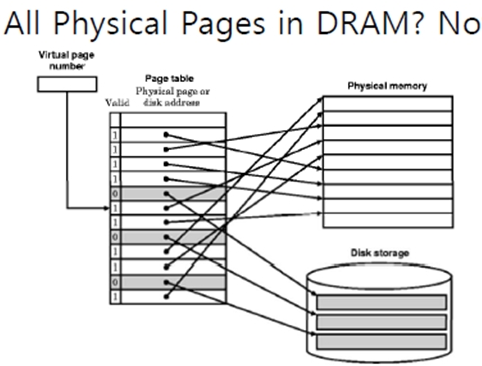
ㅇ 프로세스는 자신만의 가상 주소 공간을 가짐(logical memory)

ㅇ A 프로세스가 0x12345678 주소에 무엇인가를 저장하였지만,

B 프로세스 역시 0x12345678 주소에 무엇인가를 저장할 수 있으며, 이 주소들은 완전히 독립

2. 물리메모리

ㅇ 실제 물리적 H/W에 장착된 메모리 (physical memory)



3. page

ㅇ 가상 메모리를 사용하는 최소 크기 단위이다.

ㅇ 최근의 윈도우 운영체제,리눅스에서 4096 (4KB)의 페이지 크기사용

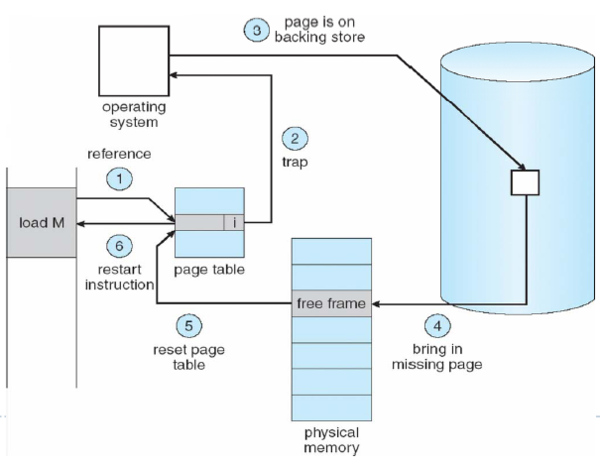
ㅇ 데이터의 크기가 모두 제각각이면 fragmentation이 발생하여 활용율이 떨어짐

4. swap

ㅇ page에 할당할 물리메모리가 부족한 경우 page중 일부를 디스크에 저장(swap-out)

ㅇ 프로세스가 swap-out된 page를 참조할 경우 page table에 없으므로 page fault가 발생하고 os가 다시 디스크에서 읽어 page table과 TLB 업데이트

(swap-in)



5. page table

ㅇ 페이지 테이블은 가상 주소와 물리 메모리 주소의 매핑 테이블이며, 개별 매핑 데이터는 Page-Table-Entry(PTE)라고 함

ㅇ 페이지 테이블은 프로세스마다 하나씩 존재하게 되며, 메인 메모리 (RAM)에 상주하게 된다.

ㅇ 페이지 테이블은 가상 주소의 페이지와 물리 메모리 프레임간의 매핑 정보를 들고 있다. 그리고, 다음과 같은 부가 정보들을 포함한다.

1)reference bit (accessed bit)

2)valid bit (present bit)

3)dirty bit (modified bit)

4)process ID information

6 .MMU

ㅇ MMU(Memory Management Unit)는 최근에 맵핑된 페이지 테이블의 정보들을 TLB라는 연관 캐쉬에 저장 (TLB : Translation Lookaside Buffer)

ㅇ TLB는 자주 쓰는 페이지 테이블의 주소를 저장해 둠으로써, translation의 속도를 향상시켜 주는 역할

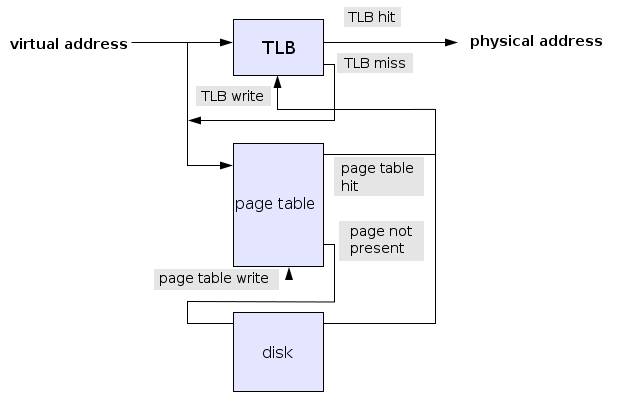
ㅇ virtual-> physical로 변환방식

1) 가상 주소가 물리 메모리 주소로 변환되어야 할 때, 먼저 TLB를 검색한다.

TLB에서 검색이 성공하면 (TLB hit), 즉시 물리 메모리 주소를 반환하며, 프로세스는 해당 주소로 접근이 가능하다.

2) 만약, TLB에서 검색이 실패하면 (TLB miss), 통상적으로 핸들러는 해당 가상 주소에 맵핑된 물리 메모리 주소가 있는지 페이지 테이블을 검색하게 된다.

3) 페이지 테이블에서 일치하는 맵핑 데이터를 찾았다면, 해당 정보를 TLB에 업데이트하고, 1) 번 과정으로 돌아가 해당 가상 주소의 물리 메모리 주소로의 요청을 재개하게 된다.



7 .MMU

ㅇ MMU(Memory Management Unit)는 최근에 맵핑된 페이지 테이블의 정보들을 TLB라는 연관 캐쉬에 저장 (TLB : Translation Lookaside Buffer)

ㅇ TLB는 자주 쓰는 페이지 테이블의 주소를 저장해 둠으로써, translation의 속도를 향상시켜 주는 역할

ㅇ virtual-> physical로 변환방식

8 IOMMU

ㅇ virtual pci address를 physical pci address로 변경하는 역할

ㅇ IOMMU를 활성화하기 위해서는 VT-d가 BIOS에서 활성화 되어야 함

ㅇ