Page? Context switching? Dual-mode?

MMU : 가상메모리와 물리메모리를 맵핑 해주는 역할을 함.

: 가상메모리를 사용하면 물리적으로 떨어져 있는 메모리를 연속적으로 사용할 수 있고, 여러 프로세스가 논리적으로 같은 메모리를 사용하게 만들 수 있음.

MMU가 맵핑을 하려면 Page table로 가서 정보를 가져와야 하는데 이건 Memory access라 느림(cpu에 비해서) 그래서 TBL라는 Cache를 두어서 빠르게 수행함.

Page (memory page, virtual page) : 가상메모리의 고정 길이의 블록이고 페이지 테이블에 있다. 가상 메모리의 최소 데이터 저장 단위이다.

* No external fragmentation
* But internal fragmentation

Page frame : 피지컬 메모리의 최소 데이터 단위

메인 메모리와 하드의 페이지 간 전송을 Paging, swapping

Page size vs Page table size : 페이지 크기가 더 작으면 더 많은 페이지를 사용하므로 페이지 테이블이 커야함

Page size vs TLB : 페이지 크기가 크면 TLB에 더 많은 페이지 정보를 담을 수 있으니까 TLB miss가 줄어듬

Page fault : 프로세스가 MMU에 멥핑 되지 않은 페이지에 접근할 때 발생

Minor : page table에 없어서 메모리에서 가져오는 경우

Major : 메모리에 있던 page가 스왑핑으로 디스크로 가서 메모리에 올릴 때 disk/io가 필요한경우

Swapping : 메모리의 프로세스를 디스크로 옮기는거 (공간 부족이나 정책등에 의해서)

== page out

Segmentation : page처럼 메모리를 쪼개는데 그 길이가 다양하게 -> no internal fragm

Fragmentation : 메모리에 프로세스를 올렸다 내렸다 하면서 hole이 생긴 것.

* Internal = page 크기보다 프로세스가 차지하는 메모리가 작아서 남는 공간
* External = 메모리에 부분부분 저장되서 큰놈이 못올라가는거