

21: swapping mechanism

Main question: 가상 메모리 어케씀?

Swap space

- 가장 먼저 할 일: 디스크에 공간 확보

메모리가 참조되는 과정

- 가상 주소에서 VPN 추출
- TLB에 정보 있는지 확인(TLB hit)
- 없다면(TLB miss) 페이지 테이블 메모리 주소를 파악, VPN을 인덱스로 하여 PTE 추출
- TLB에 이거 탑재

Present Bit

- 하드웨어가 PTE에서 해당 페이지가 물리 메모리에 존재하지 않는다는 것을 표현해야함
- 이때 사용하는 거시 present bit
- 1이면 존재, 0이면 없음
- 이때 물리 메모리에 존재하지 않는 페이지를 접근하는 행위를 page fault라 함

Page Fault

- page fault handler가 이것에 대한 업무를 담당
- 요청된 페이지가 메모리에 없고 디스크에 스왑되었다면 → 해당 페이지를 메모리에 스왑
 - 디스크 I/O 완료 시 PTE의 PFN 값을 탑재한 페이지의 메모리 위치로 갱신
 - page fault 재실행
 - 이때 TLB Miss가 나옴 → TLB 갱신!
- I/O 작업 중 다른 프로세스 가능!(느리므로 ..)

Page replace

- page-in에서 메모리가 충분하지 않다면 replace를 실행해줘야 함
- 여유 공간의 최대/최소값을 활용해 교체 알고리즘을 작동시킴
 1. 여유공간 < 최소값
 - a. 여유공간 확보용 background thread 실행
 - b. 확보되면 슬립
- 한번에 여럿을 묶어서 스왑하면 효율이 오름
 - 디스크의 탐색/회전 지연 오버헤드가 줄어듦

page fault 처리

```
1 PFN = FindFreePhysicalPage()
2 if (PFN == -1) // 비어있는 페이지 못 찾을
3     PFN = EvictPage() // 교체 알고리즘 실행
4     \gndx{DiskRead}(\gndx{PTE, DiskAddr, pfn}) // 대기 (I/O를 기다리기)
5     PTE.present = True // 존재한다고 페이지 테이블에 갱신
6     PTE.PFN = PFN // 비트와 변환 (PFN)
7     RetryInstruction() // 명령어 재시도
```

〈그림 24.3〉 페이지 오류 제어 흐름의 알고리즘 (소프트웨어)

Hardware 기반 TLB

- 하드웨어가 페이지 테이블을 검색, 정보를 찾음

software 기반 TLB

- 운영체제가 처리