Virtual Memory

지난시간 : 물리적인 메모리의 주소 변환은 운영체제가 관여하지 않는다

virtual memory의 경우 관여한다

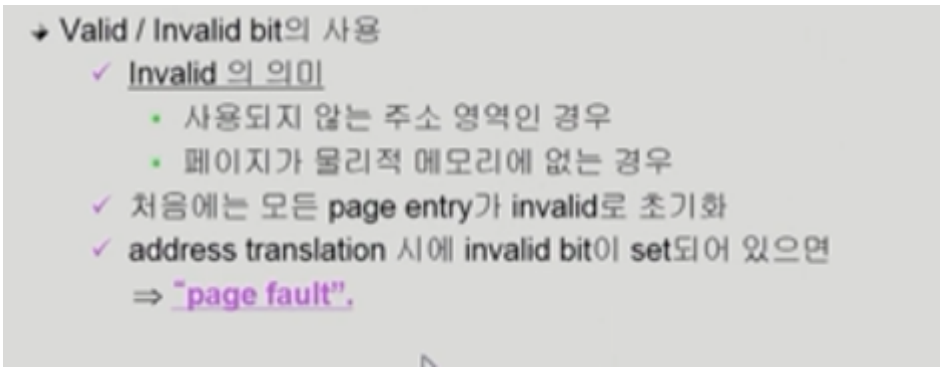
Demand Paging

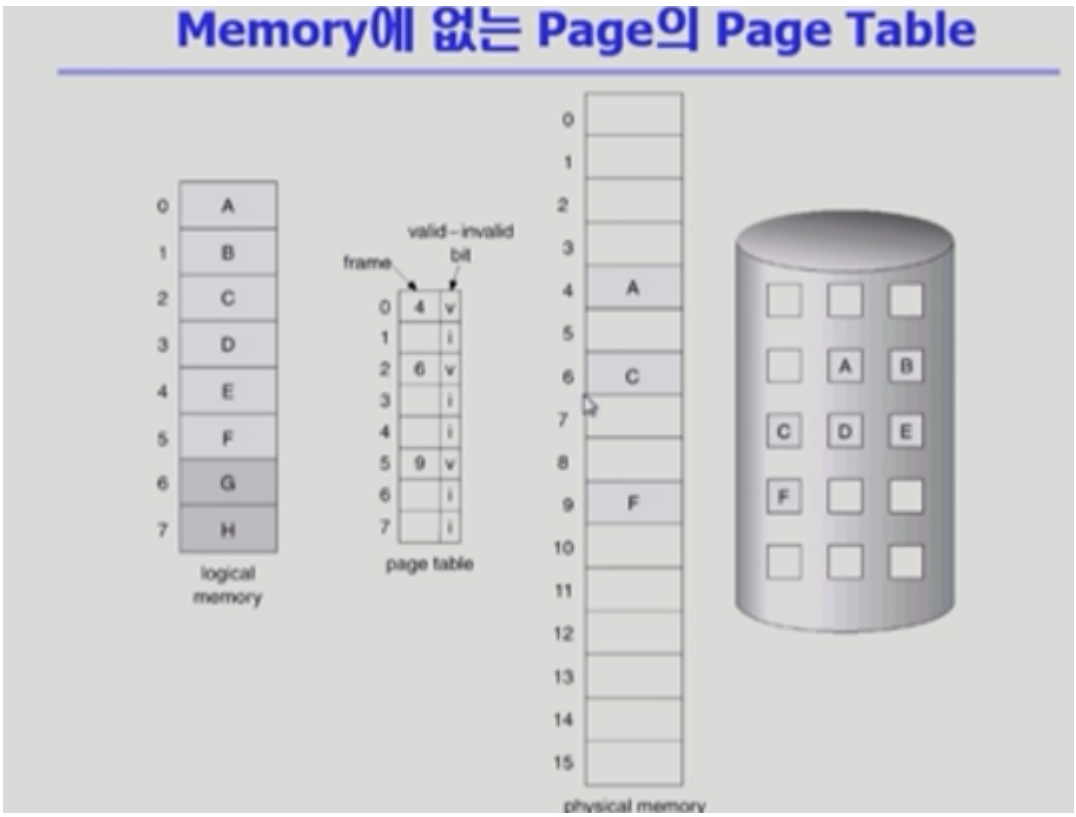
요청이 있을때만 메모리에 페이지를 올려 놓는다

=> I/O 양이 줄어듬

=> Memory 사용량 감소

=> 빠른 응답 시간

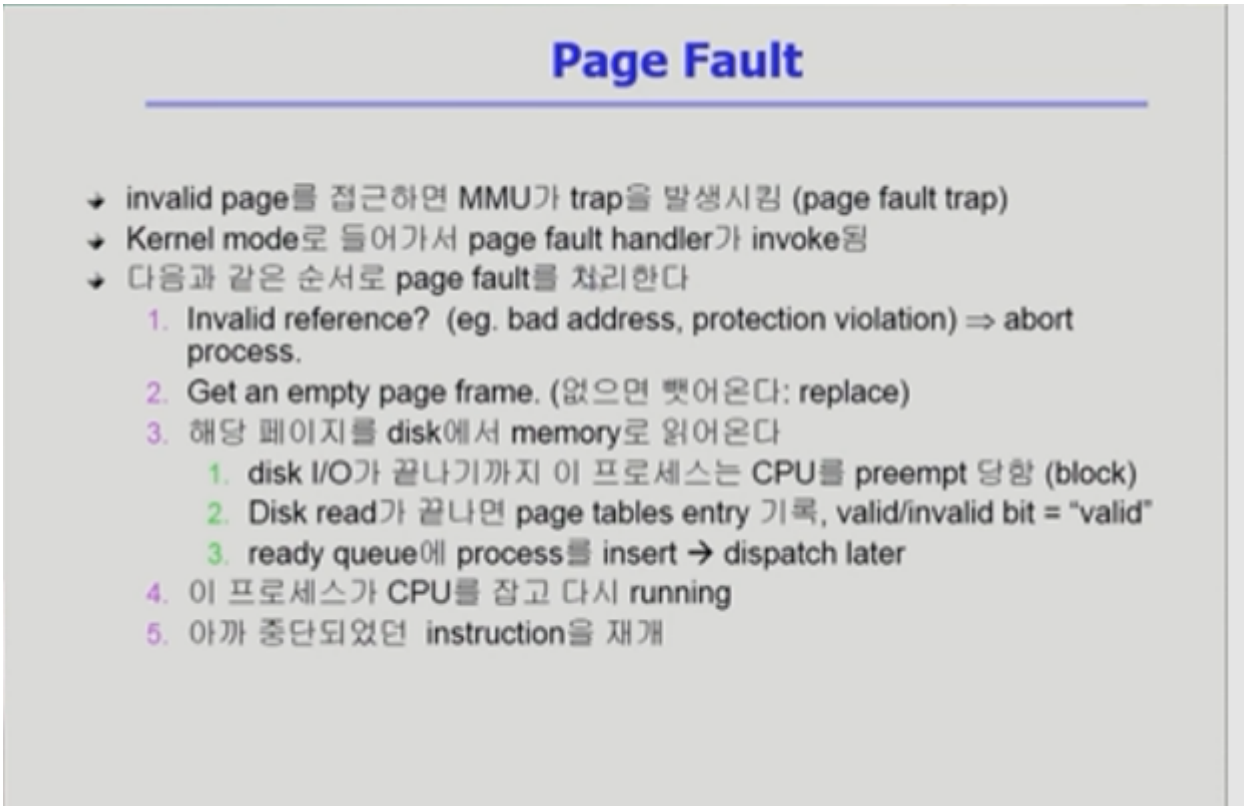




- page fault : 요청한 페이지가 메모리에 없는 경우

=> cpu가 운영체제에게 넘어감

=> 그럼 운영체제가 fault난 페이지를 메모리에 올림



Page replacement

빈 페이지가 없는 경우에

메모리에서 기존 페이지를 쫓아내고 새로운 메모리로 대체시켜줌 (OS의 일)

Replacement Algorithm

=> 어떤 것을 Disk로 쫓아낼지 정하는 알고리즘

=> page fault rate를 최소화 하는 것이 목표

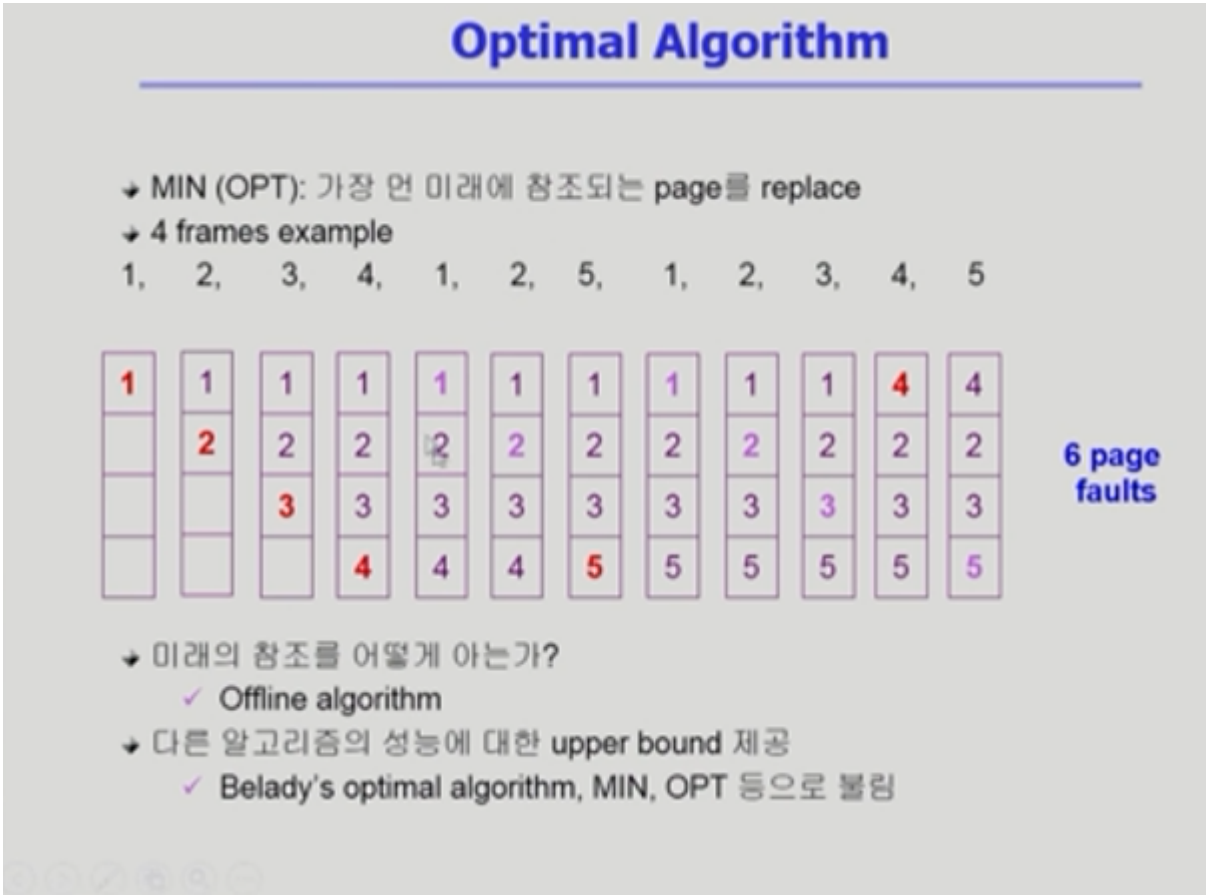
Optimal Algorithm

page 순서를 미리 알고 있다고 가정

=> 가장 먼 미래에 참조되는 Page를 쫓아냄

=> 어떤 알고리즘도 이보다 더 적은 replacement를 할 수 없음

=> 그래서 새로만든 알고리즘의 성능 척도 측정용



FIFO Alogorithm

: 먼저 들어온 것을 먼저 내 쫓음

** FiFo Anomally ** :

메모리 프레임을 늘렸으나 성능이 별로임

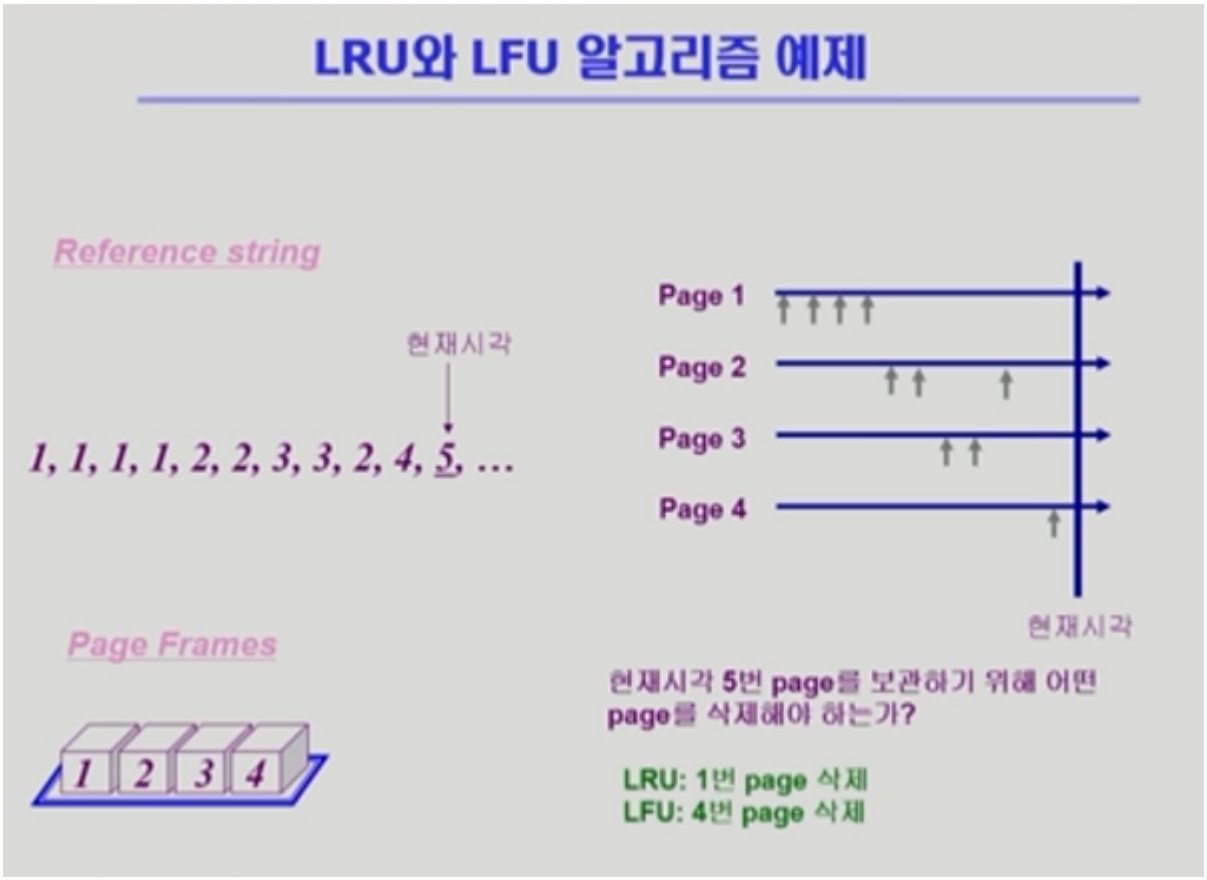
LRU Algorithm (Least Recently Used

: 가장 오래전에 참조된 것을 쫓아냄

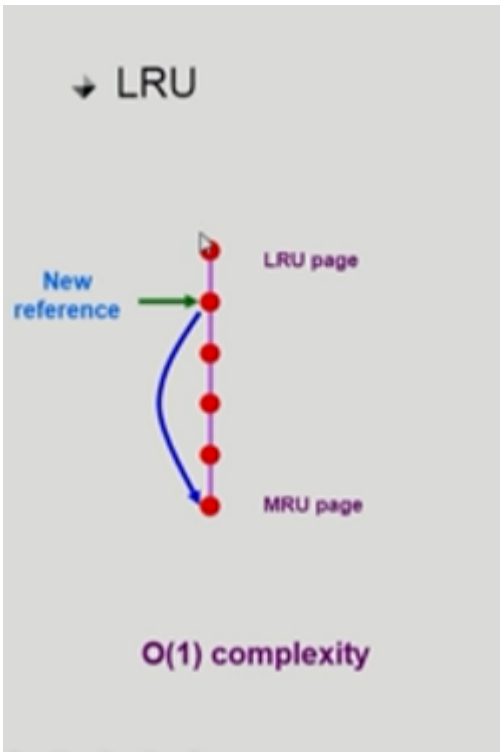
LFU Algorithm (Least Frequently Used

: 참조 횟수가 가장 적은 페이지를 지움

참조횟수가 동률일 경우 임의선정



- LRU

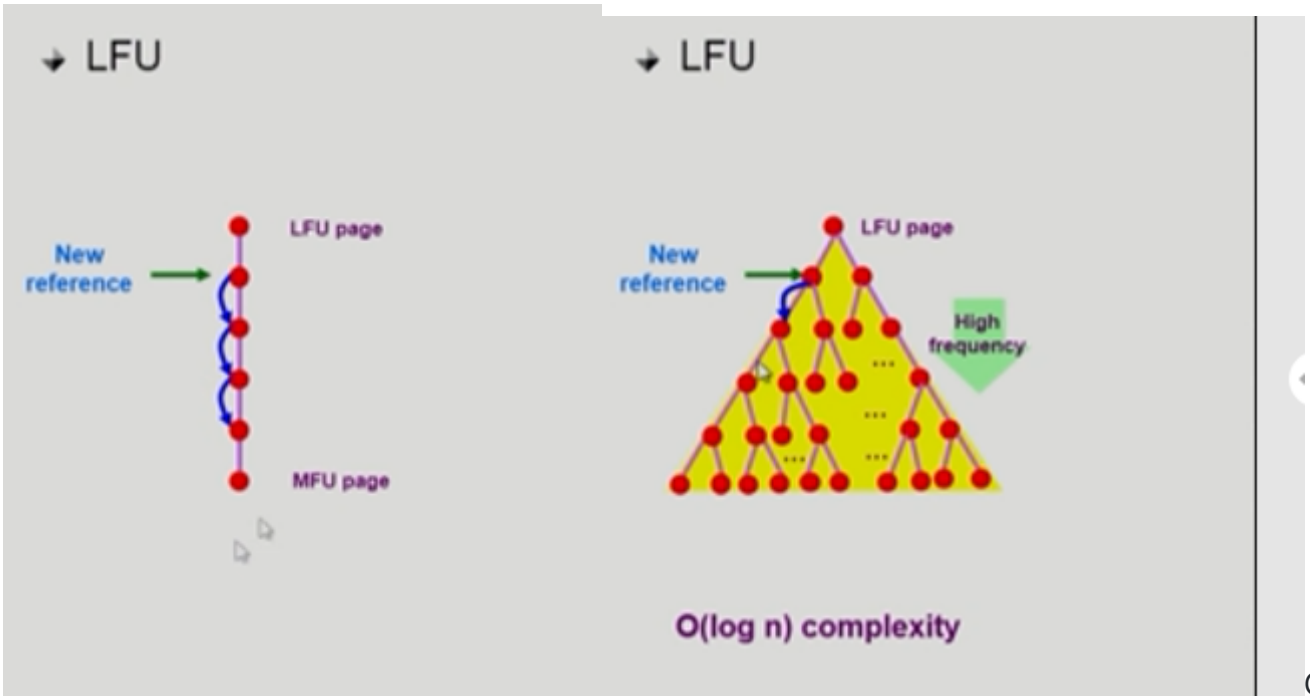


어떤 페이지가 참조 될 때마다 맨 아래 메달고

쫓아낼 때는 제일 위에거 쫓아내고

이런 방식으로 구현

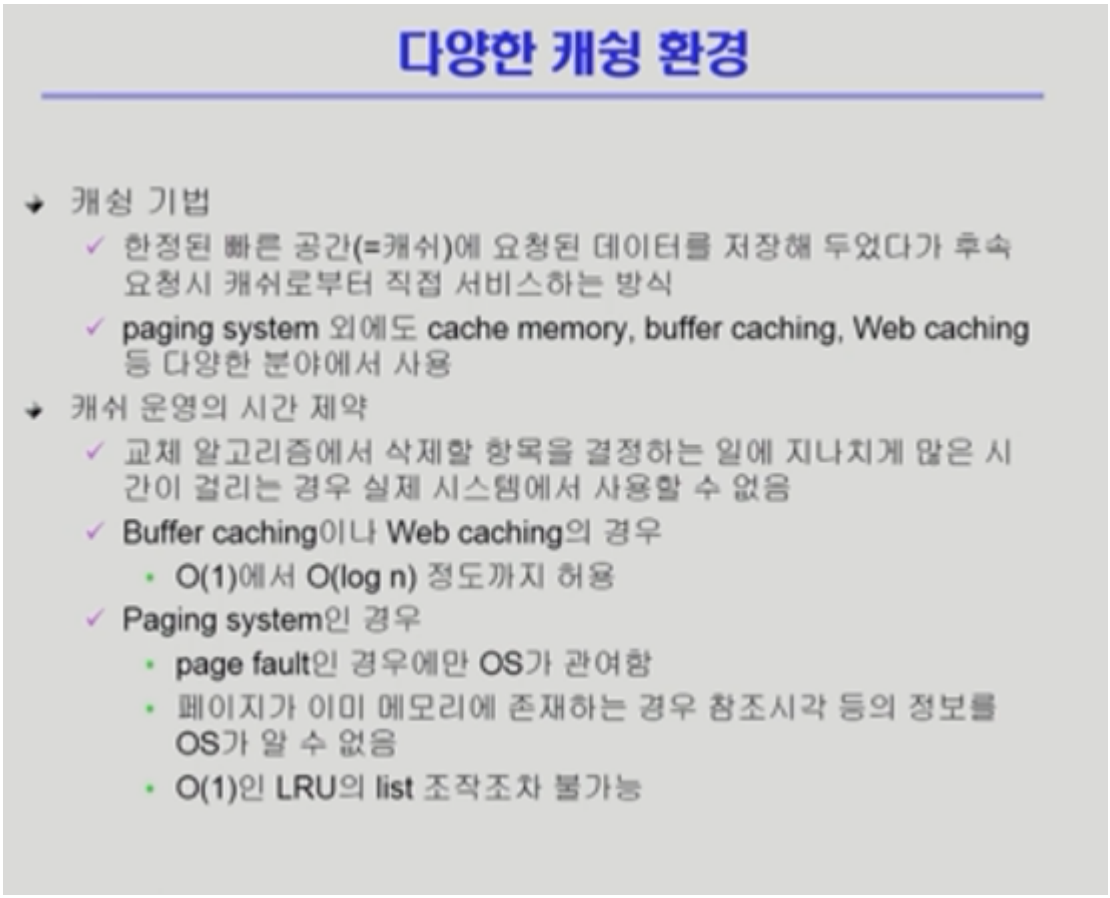
- LFU
- 참조회수 1회 늘어난걸로 바로 다른 페이지들과 비교 가능한 것이 아님
- Heap 이용해서 이진 트리를 구성
- 참조횟수가 바뀔 때 아래위로 비교하고 위치 조정



O(n)

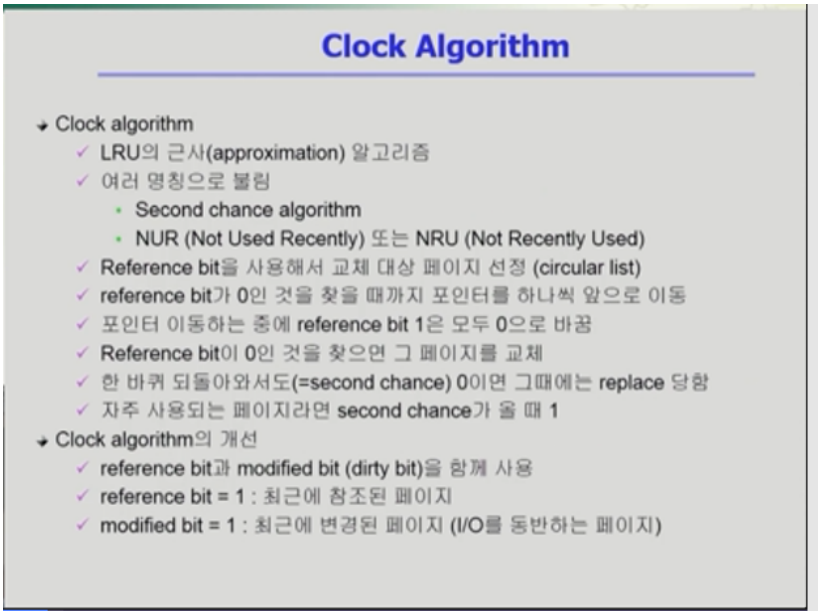
= log n

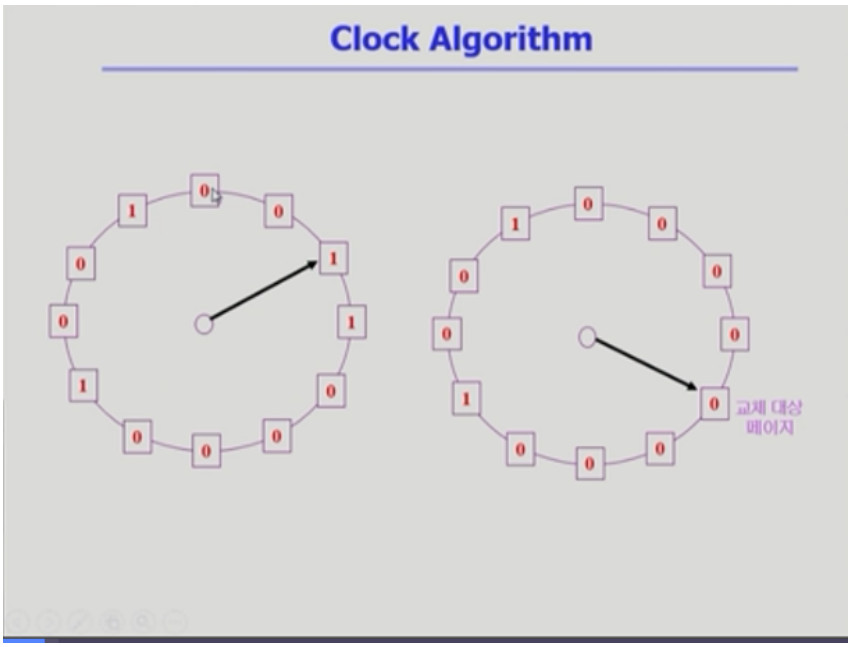
캐쉬



!!! page fault 의 경우에만 운영체제에게 Cpu가 넘어옴

Clock Algorithm





각각의 사각형은 페이지 프레임

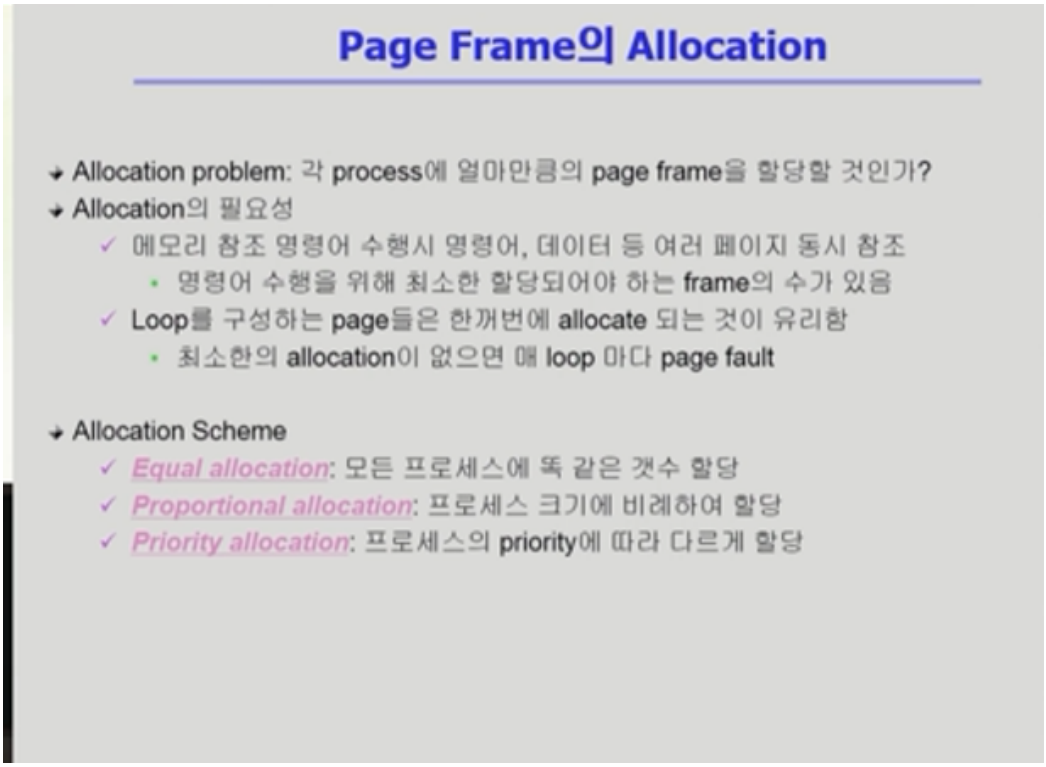
Reference bit : 페이지가 참조될 때 1로 체크해줌, 하드웨어의 일

Reference bit = 1 이라는 뜻 == 최근에 참조됨

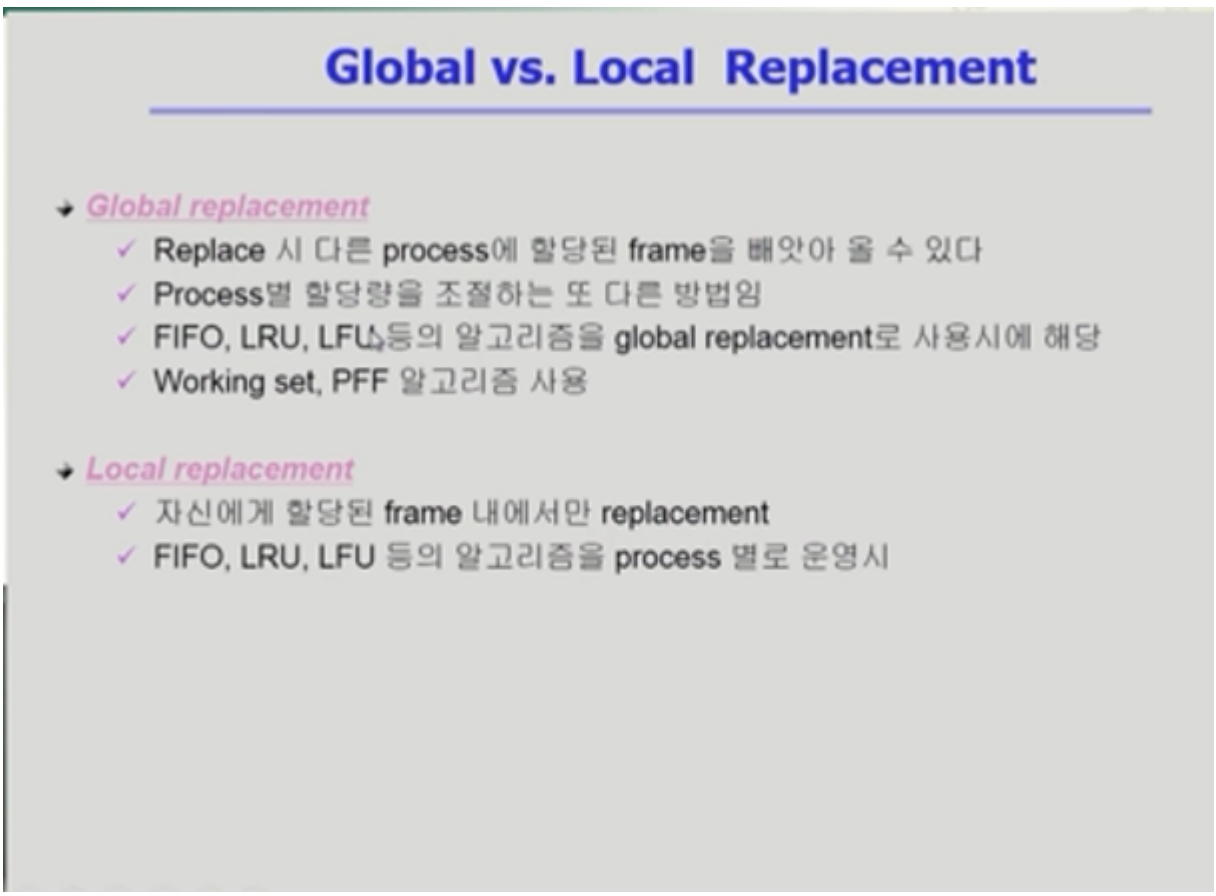
modified bit :

write 참조 일 때 내용이 수정 되었다는 뜻이므로 Disk에 변경사항 써주고 쫓아냄

Page Frame의 Allocation



- Equal allocation : 모두 균등 할당
- Proportional allocation : 프로그램 크기에 비례해서 페이지 할당



Global repalcement

replacement 알고리즘이 알아서 할당 효과를 내게 함

Local replacement

프로그램들에게 미리 메모리를 할당함

새로운 페이지가 필요할 때 자신에게 할당된 페이지를 쫓아냄