

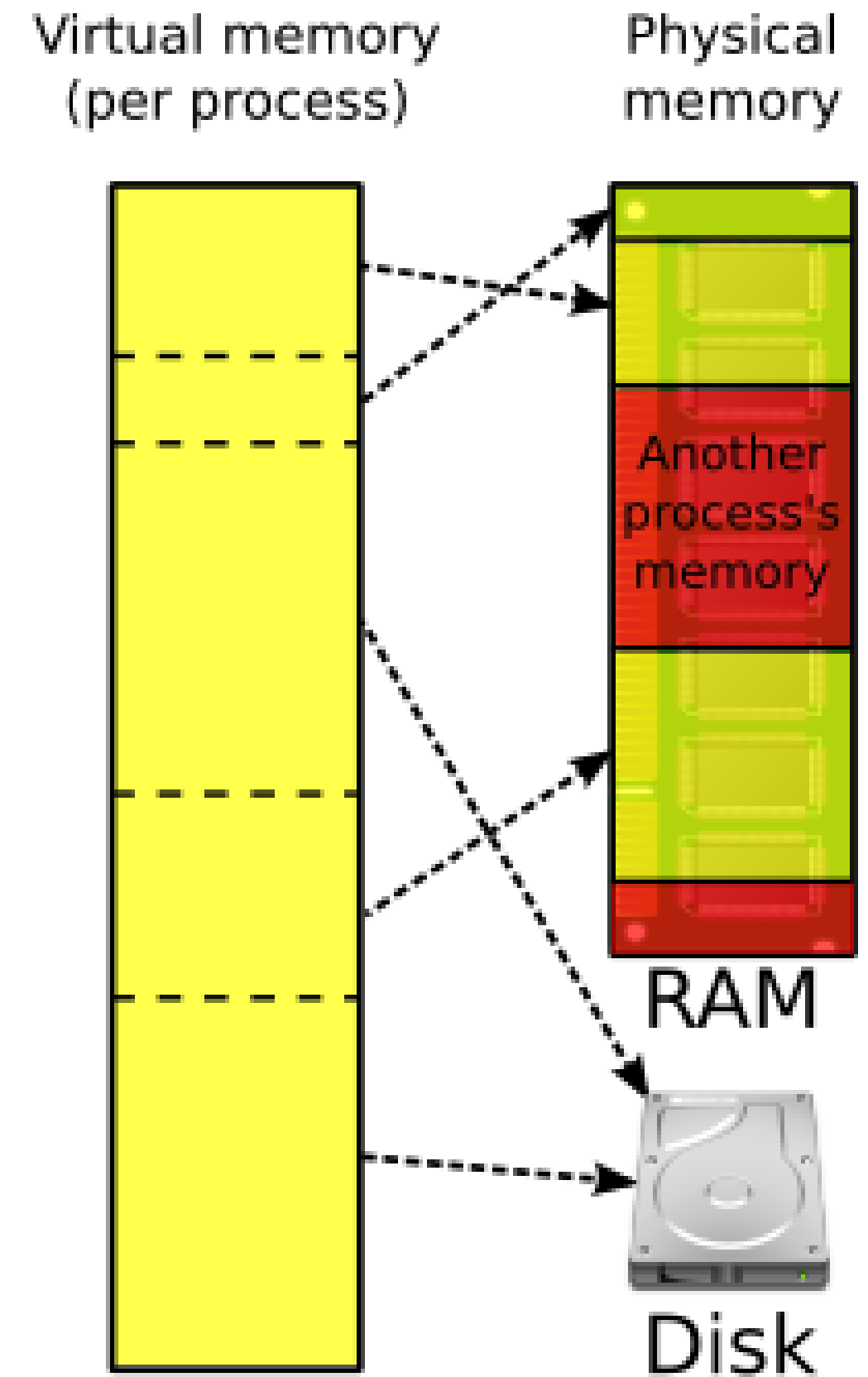
페이지 교체 알고리즘 프레젠테이션

가상기억장치

- 보조기억 장치의 일부를 주기억장치처럼 사용
- 용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용하는 기법

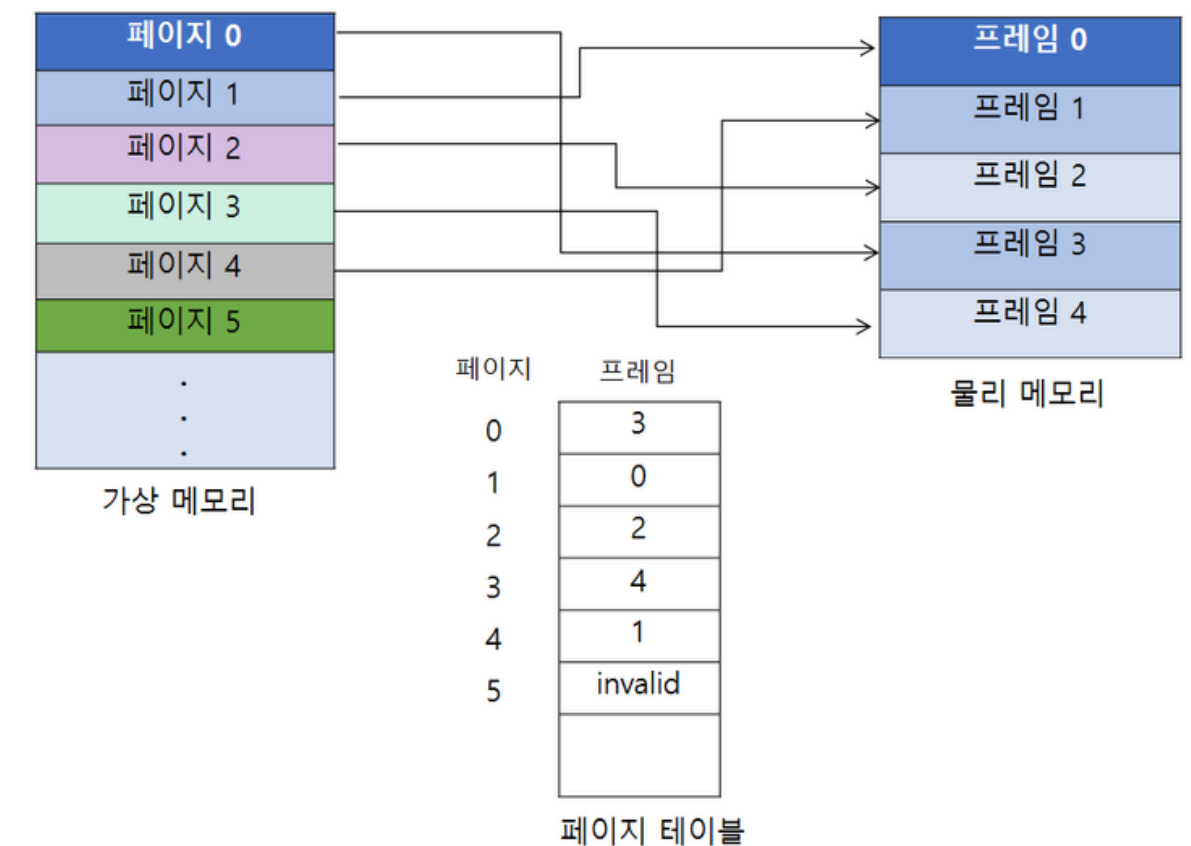
가상기억장치의 구현 방법

- 1.페이징 기법
- 2.세그먼테이션 기법



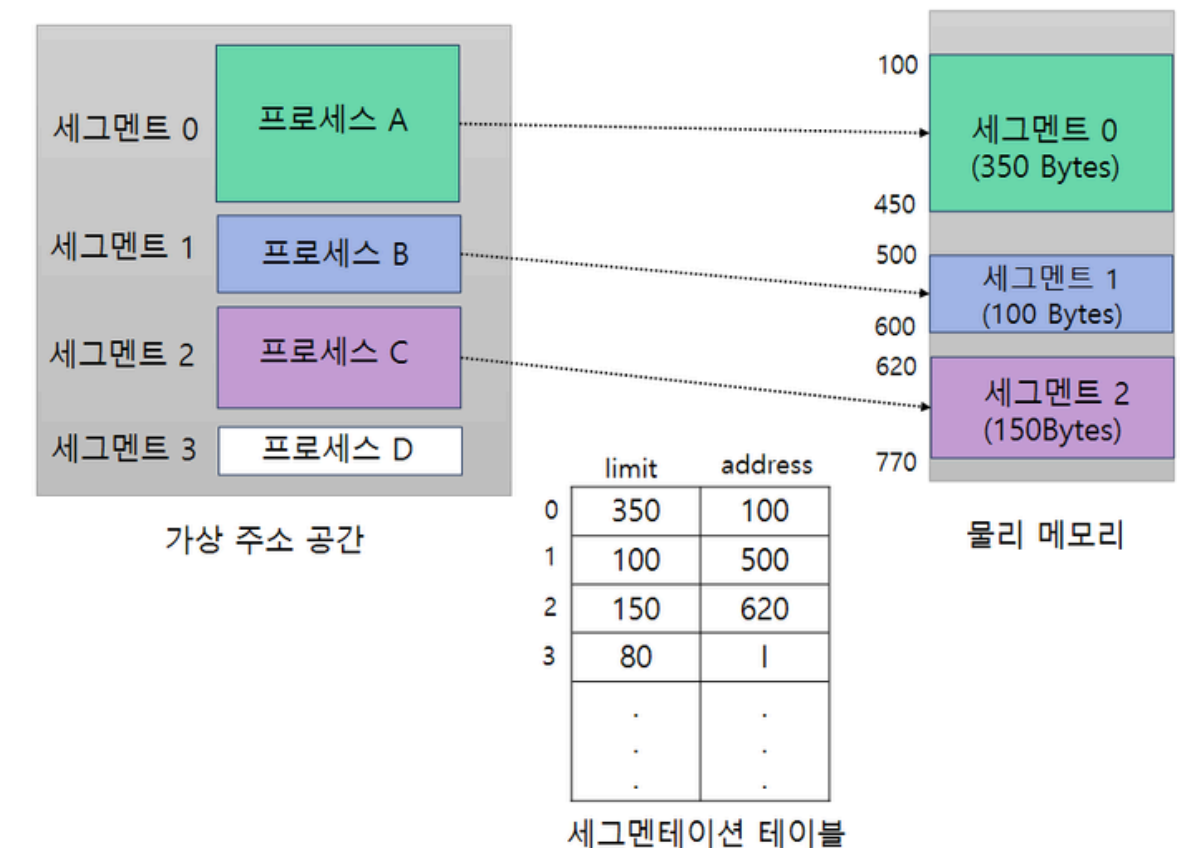
페이징 기법

- 가상기억장치에 보관되어 있는 프로그램과 주 기억장치의 영역을 동일한 크기로 나눈 후, 나눠진 프로그램을 동일하게 나눠진 주기억장치의 영역에 적재시켜 실행하는 기법
- 프로그램을 일정한 크기로 나눈 단위를 페이지라고 하고, 페이지 크기로 일정하게 나누어진 주기억장치의 단위를 페이지 프레임이라고 함



세그먼테이션 기법

- 가상기억장치에 보관되어 있는 프로그램을 다양한 크기의 논리적인 단위로 나눈 후 주기억장치에 적제시켜 실행시키는 기법
- 프로그램을 배열이나 함수 등과 같은 논리적인 크기로 나눈 단위를 세그먼트라고 하며, 각 세그먼트는 고유한 이름과 크기를 갖음



페이지 교체 알고리즘

아시겠죠 ?

- 페이지 부재가 발생하면 가상기억장치에서 필요한 페이지를 찾아 주 기억장치에 적재해야 함
- 이 때, 주기억장치의 모든 페이지 프레임이 사용중이면 어떤 페이지 프레임을 선택하여 교체할 것인지 결정해야 함

(페이지 부재: CPU가 참조하려는 페이지가 현재 메모리에 올라와 있지 않음)

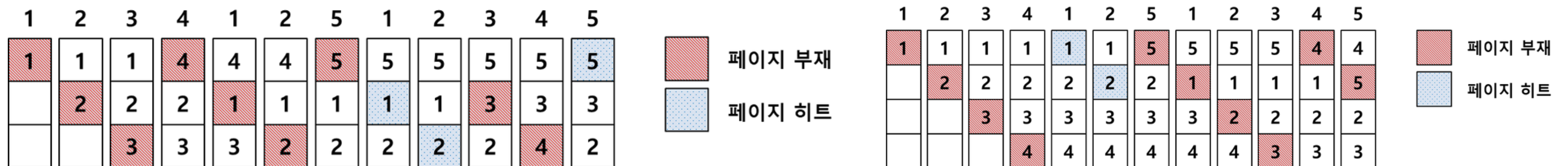


FIFO (First In First Out)

- 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법
- 페이지의 향후 참고 가능성을 고려하지 않고 물리적 메모리에 들어온 순서대로 교체하기 때문에 비효율적인 상황이 발생할 수 있음

빌레이디의 모순

- 페이지 프레임 수가 많으면 페이지 부재 수가 줄어드는 것이 일반적이지만 페이지 프레임 수를 증가시켰음에도 페이지 부재가 더 많이 발생하는 현상

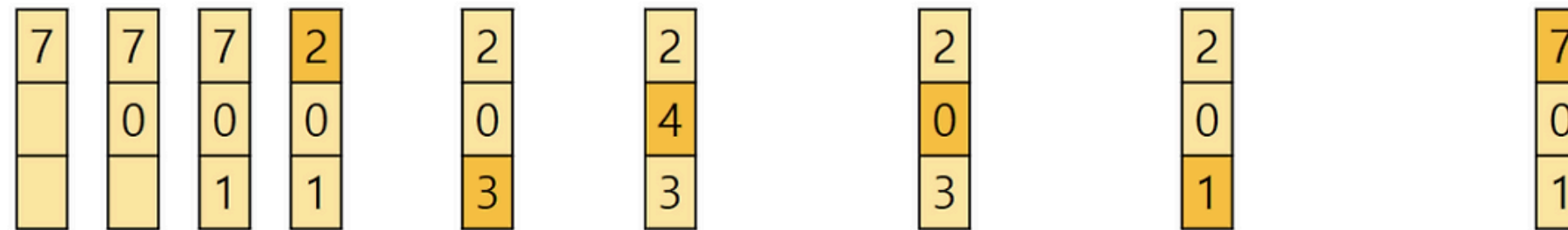


OPT (OPTimal replacement)

- 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법
- 페이지 부재 횟수가 가장 적게 발생하는 가장 효율적인 알고리즘
- 하지만, 미래에 어떤 페이지가 어떤 순서로 참조될지 미리 알고 있다는 전제하에 운영되기 때문에 현실적으로 구현이 어렵다.

참조 페이지 번호(붉은 색은 page fault)

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1




페이지 프레임


프레임의 페이지 중
1이 가장 뒤에 쓰임을
미리 알고 교체

LRU (Least Recently Used)

- 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법
- OPT의 경우 미래예측이지만, LRU는 과거를 보고 판단하기 때문에 실질적으로 사용 가능한 알고리즘
- 실제로 사용할 수 있는 페이지 교체 알고리즘에서는 가장 좋은 방법 중 하나

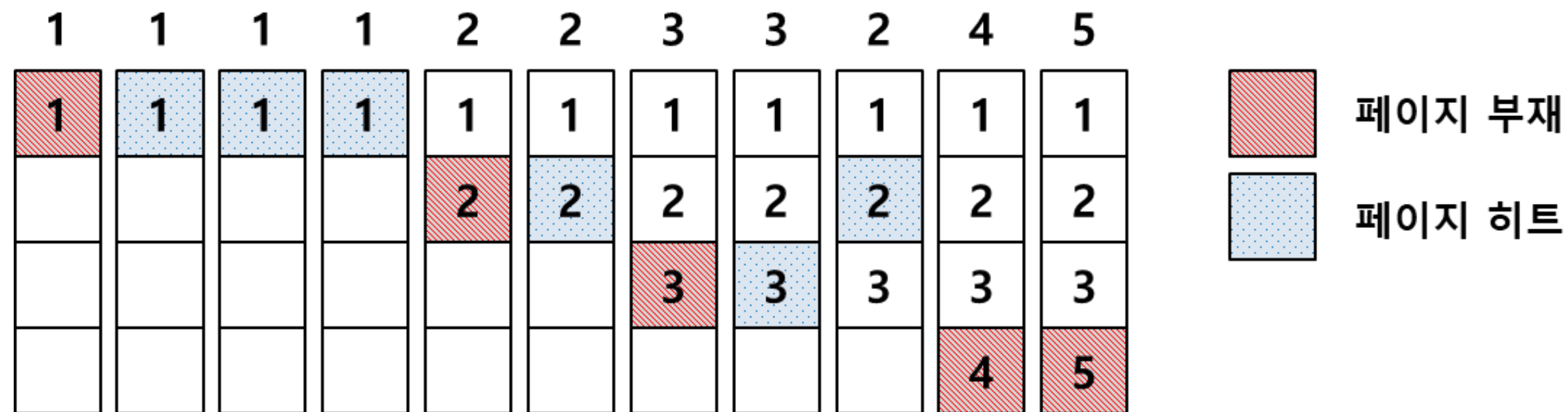
1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	1	1	1	1	1	1	1	5
	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	5	5	5	5	4	4
			4	4	4	4	4	4	3	3	3

 페이지 부재

 페이지 히트

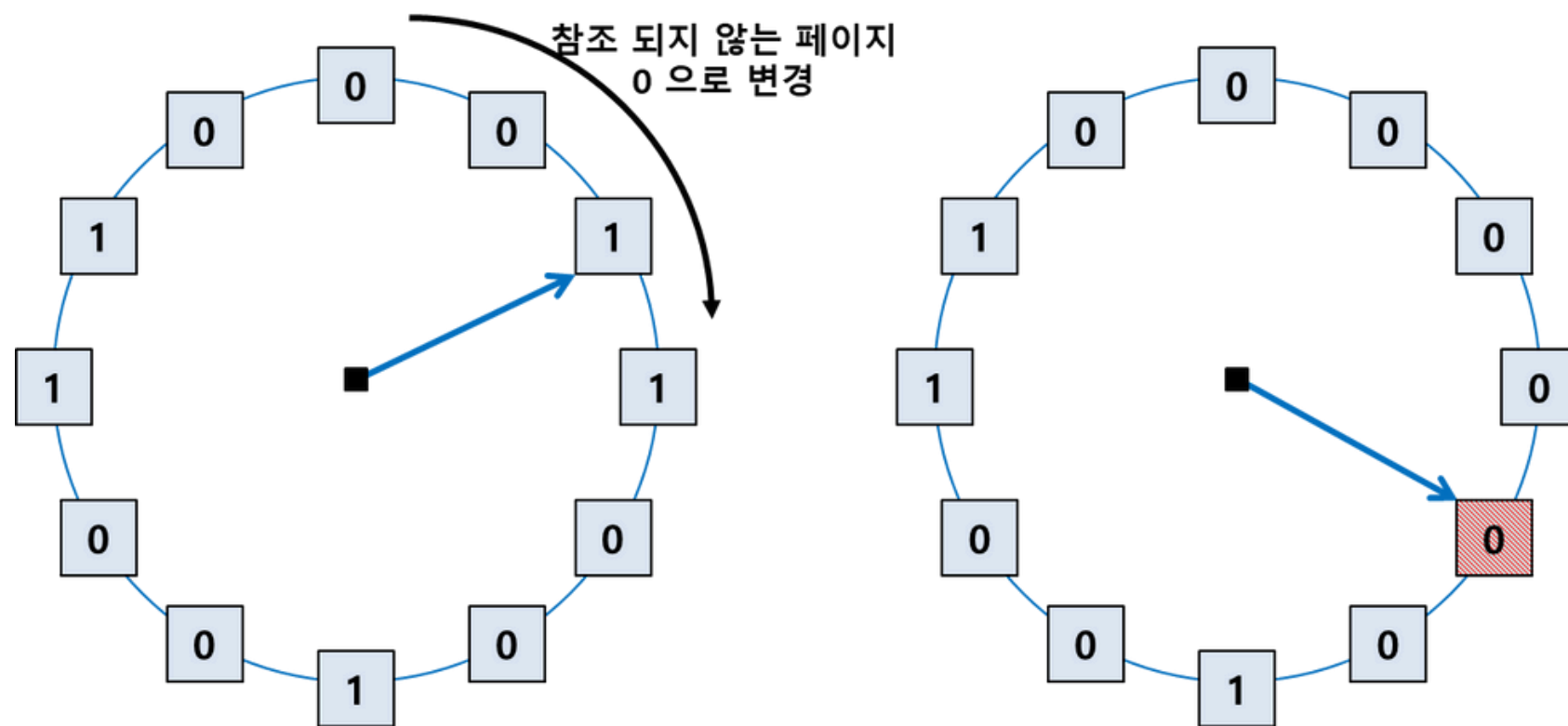
LFU (Least Frequently Used)

- 사용 빈도가 가장 적은 페이지를 교체하는 기법
- 시간에 따른 페이지 참조의 변화를 반영하지 못하고, LRU보다 구현이 복잡함
- LRU, LFU 알고리즘은 페이지의 최근 참조 시각 및 참조 횟수를 소프트웨어적으로 유지해야 하기에 알고리즘 운영에 오버헤드 발생



NUR (Not Used Recently)

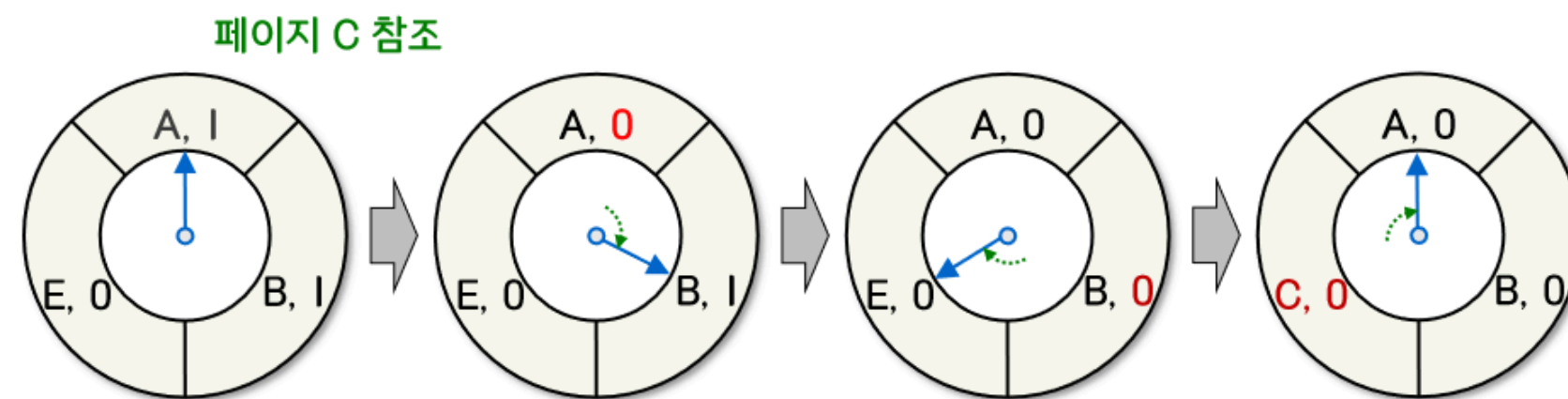
- 하드웨어적인 지원을 받아 시간적인 오버헤드를 줄인 방식
- 오랫동안 사용하지 않은 페이지 중 하나를 교체
- 참조 비트: 페이지가 참조될 때는 1, 참조되지 않았을 때는 0으로 지정
- 변형 비트: 페이지 내용이 변경되었을 때 1로 지정



참조비트	변형비트	교체순서
0	0	1
0	1	2
1	0	3
1	1	4

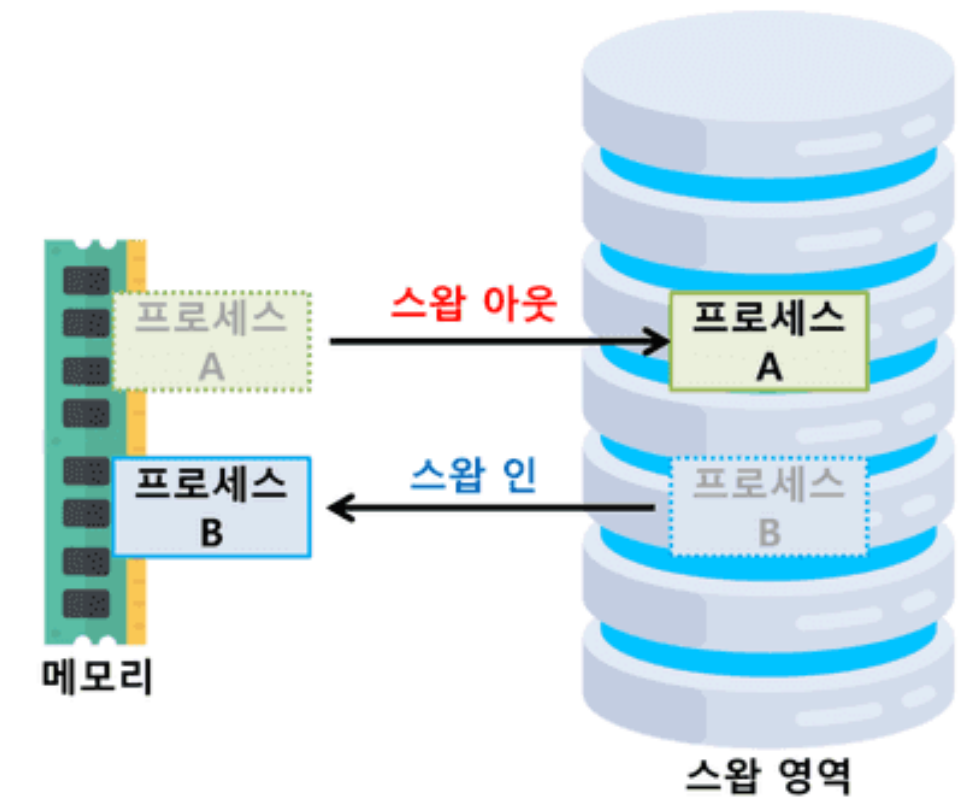
SCR (Second Chance Replacement)

- 가장 오랫동안 주기억장치에 있던 페이지 중 자주 사용되는 페이지의 교체를 방지하기 위한 것으로, 참조비트를 추가하여 FIFO 기법의 단점을 보완하는 기법
- 페이지들은 FIFO 방식으로 큐에 저장됨 (맨 앞부터 교체 대상)
- 교체 대상 페이지의 참조비트를 확인하고, 0이면 즉시 교체하지만 1이면 페이지를 큐의 뒤로 이동하고 0으로 초기화 (두 번째 기회 부여)



세그먼테이션 기법에서는 ?

- 세그먼테이션 기법은 가변적인 크기의 블록을 다루기 때문에 단순히 일부 데이터를 교체하는 것이 아니라 전체 세그먼트를 이동해야 함
- 세그먼트는 주로 스와핑 방식으로 관리
(필요한 세그먼트 전체를 디스크에서 가져오고 기존 세그먼트를 내보냄)



FINISH
GOOD!