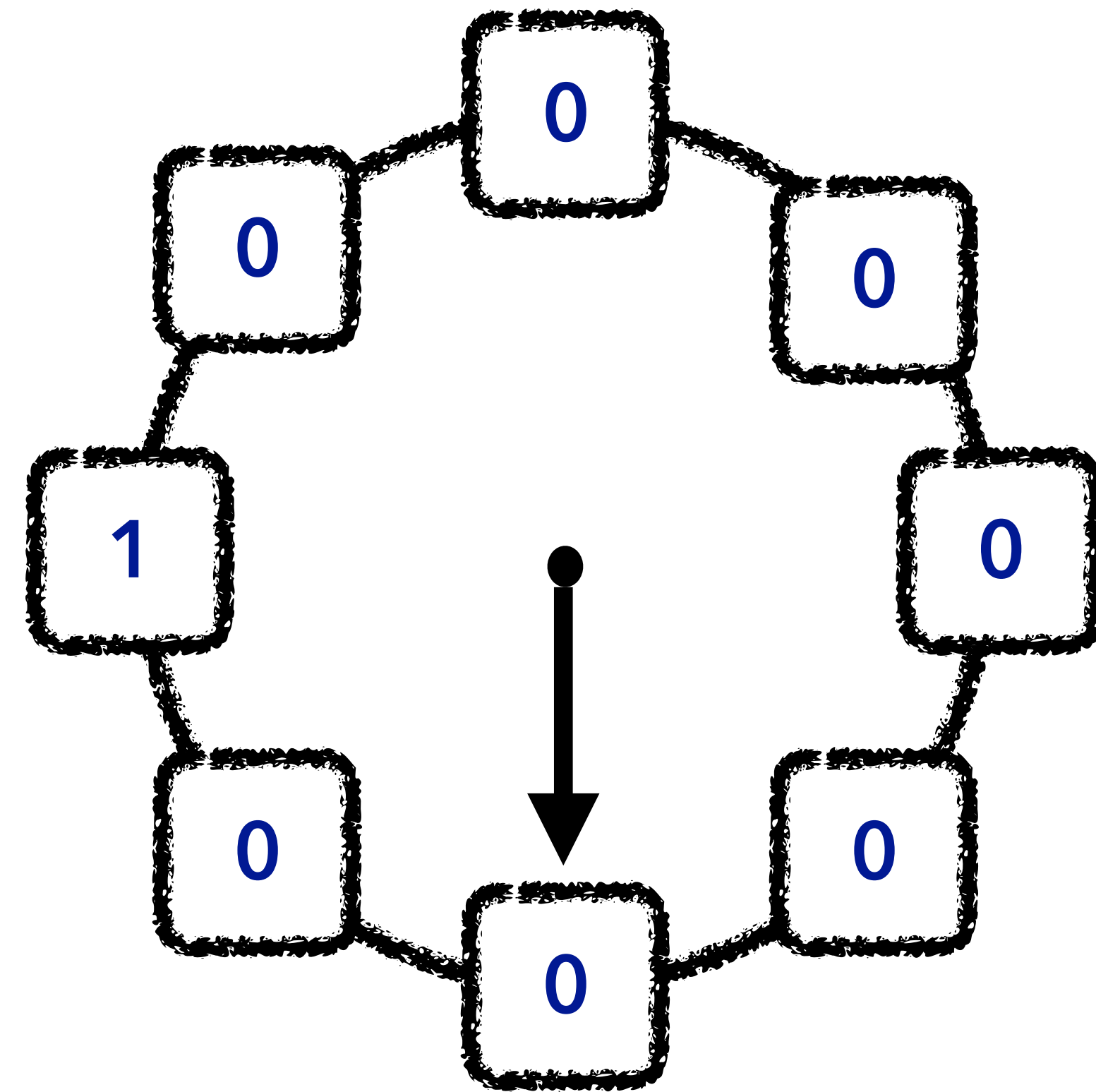
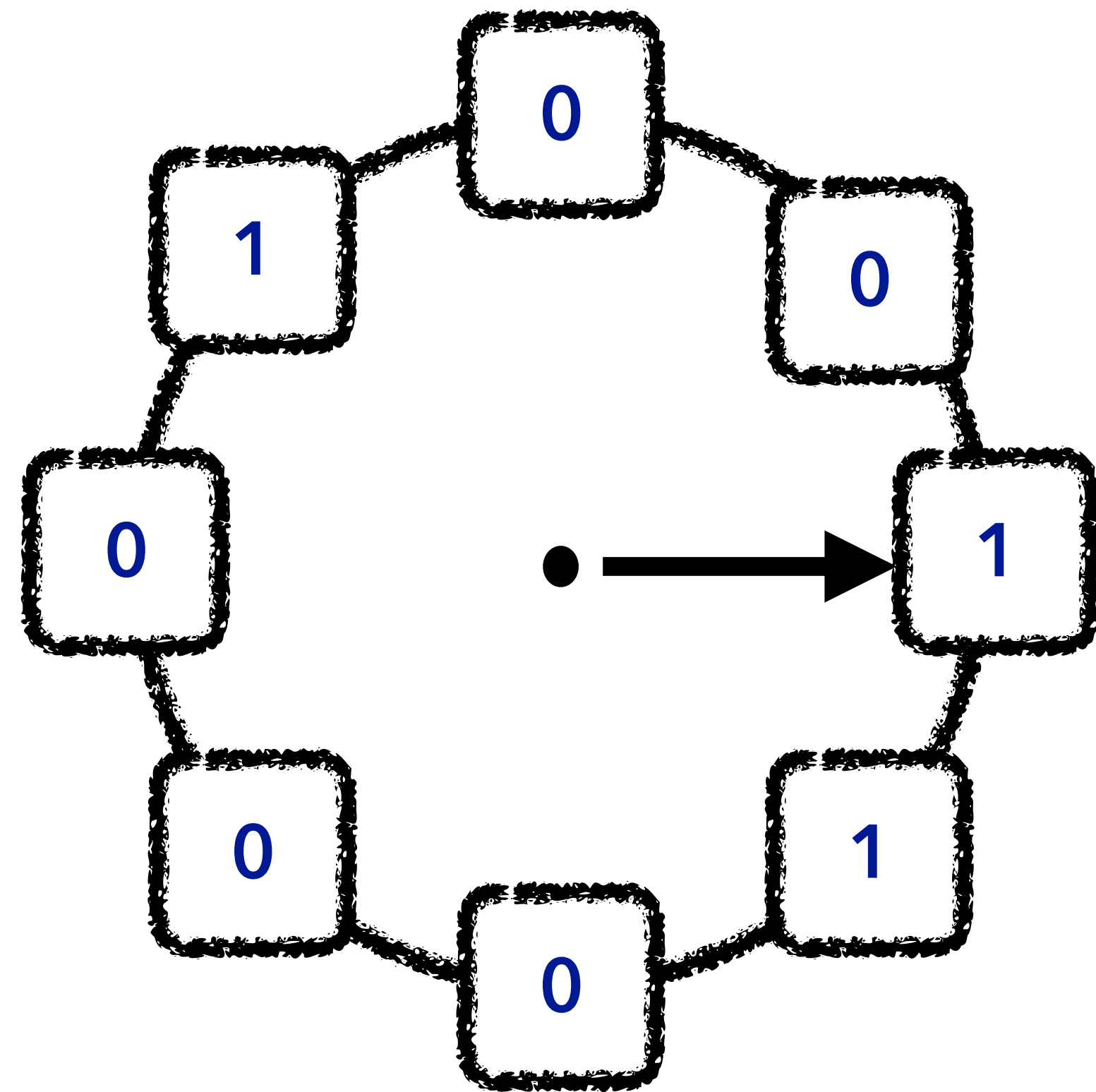


# 가상 메모리

# CLOCK ALGORITHM



교체 대상 페이지

# CLOCK ALGORITHM

---

- LRU에 가까운 알고리즘
- 여러 명칭으로 불림: Second chance algorithm, NUR(Not Used Recently), NRU(Not Recently Used)



## Clock algorithm의 개선

Reference bit과 modified bit(dirty bit)을 함께 사용

Reference bit = 1: 최근에 참조된 페이지

Modified bit = 1: 최근에 변경된 페이지 (I/O를 동반하는 페이지)

# PAGE FRAME 의 ALLOCATION

Allocation Problem: 각 프로세스에 얼마만큼의 Page Frame을 할당할 것인가?

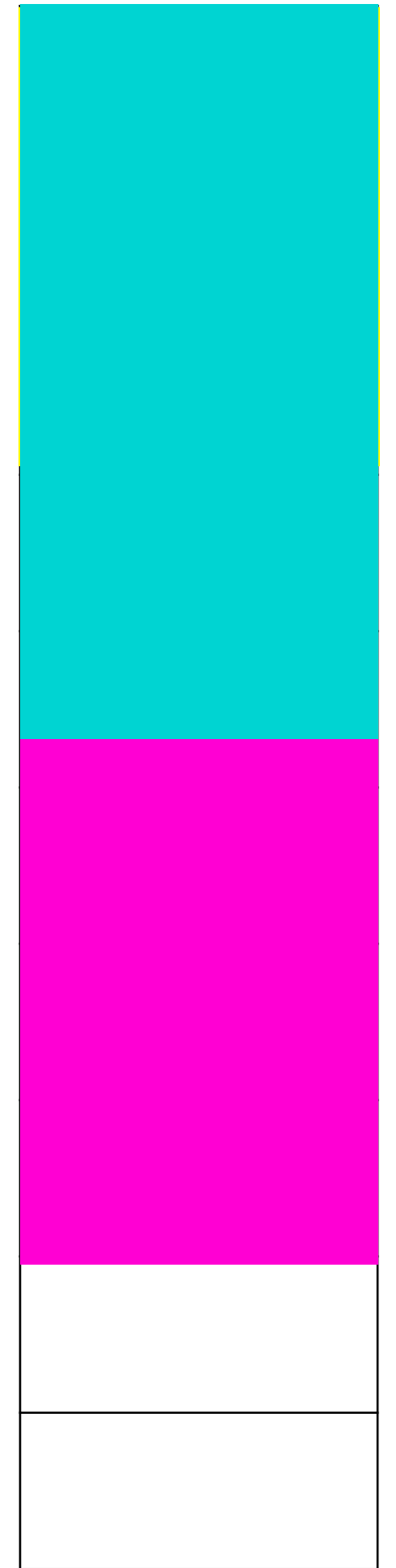
Allocation의 필요성?

메모리 참조 명령어 수행시 명령어, 데이터 등 여러 페이지 동시 참조

› 명령어 수행을 위해 최소한 할당되어야 하는 frame의 수가 있음

Allocation scheme

- Equal allocation : 모든 프로세스에 똑같은 갯수 할당
- Proportional allocation: 프로세스 크기에 비례하여 할당
- Priority allocation: 프로세스의 priority에 따라 다르게 할당



메모리

# GLOBAL VS LOCAL REPLACEMENT

---

## Global replacement

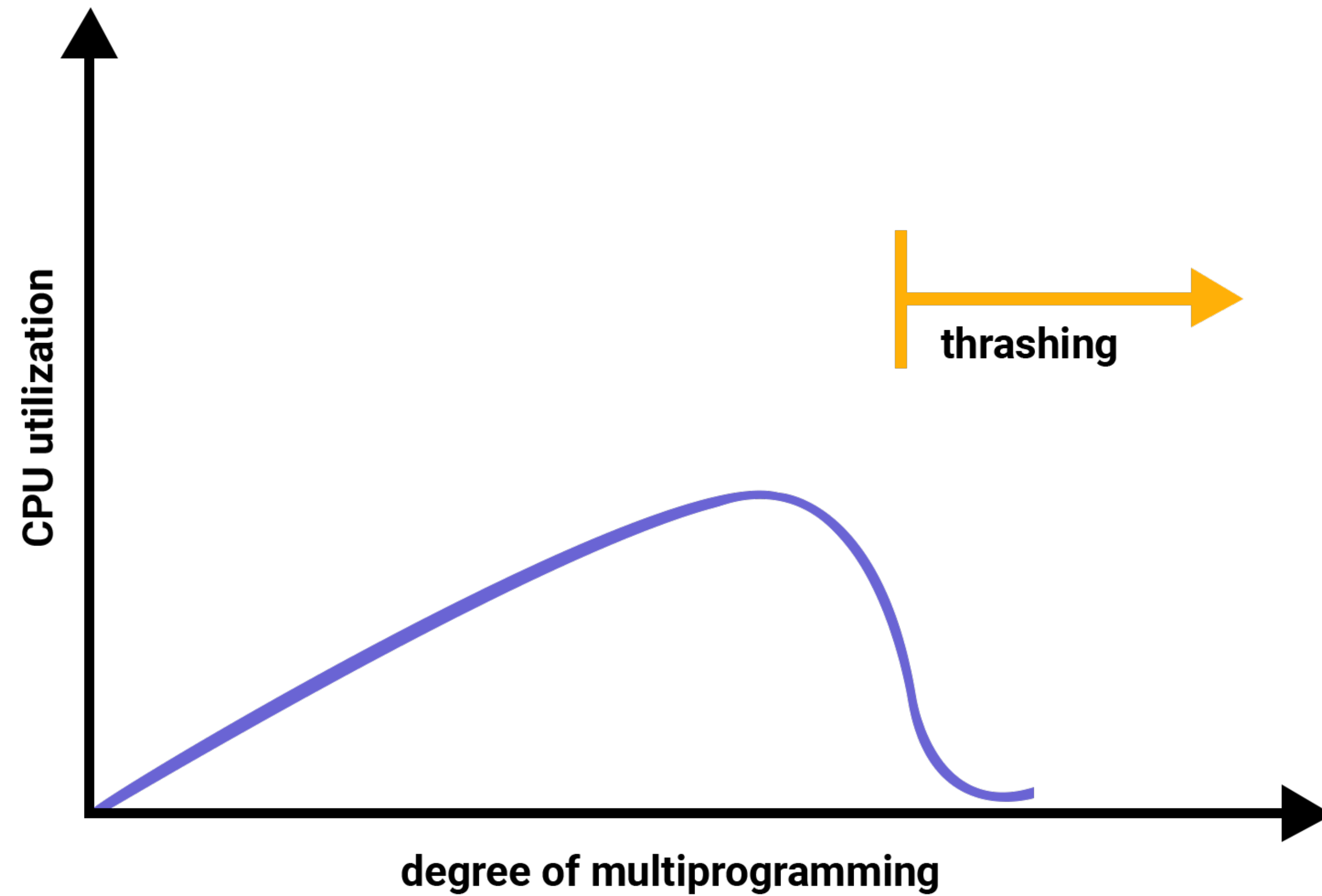
- 각 프로세스에 별도로 메모리 할당을 하지 않고 GLOBAL하게 알고리즘에 따라 페이지가 쫓겨나도록 함. (어떤 프로세스의 페이지인지 신경쓰지않음)
- FIFO, LRU, LFU등의 알고리즘을 global replacement로 사용시에 해당
- 좋은 점  
필요한 프로그램에 메모리를 많이 줄 수 있다.  
(메모리를 많이 사용하는 프로그램에게 메모리가 많이 할당할 수 있음.)
- 문제점  
특정 프로그램이 메모리를 독식 할 수 있다

## Local replacement

- 각 프로세스에 별도로 메모리 할당을 하고 자신에게 할당된 frame내에서만 replacement
- FIFO, LRU, LFU등의 알고리즘을 프로세스 별로 사용시에 해당

# THRASHING

---



# WORKING-SET MODEL

---

## Locality of reference

프로세스는 특정 시간 동안 일정 장소만을 집중적으로 참조.  
집중적으로 참조되는 해당 page들의 집합을 Locality set이라고 함.

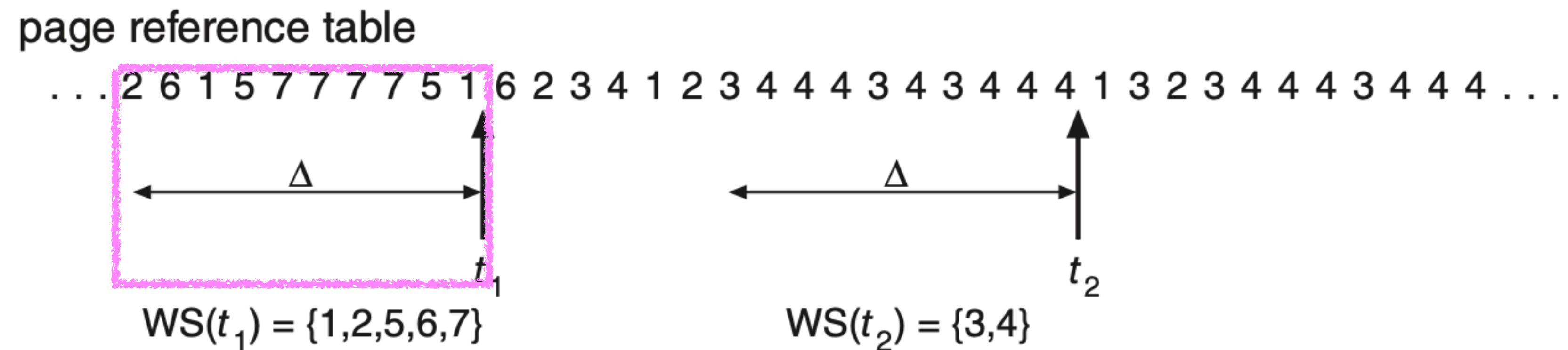
## Working-Set?

한꺼번에 메모리에 올라와 있어야 효율적으로 사용되는 페이지들의 집합



Working-Set 모델에서는 프로세스의 Working-Set 전체가 메모리에 올라와 있어야 수행되고 그렇지 않을 경우 모든 프레임을 반납한 후 Swap out(suspend)된다.

# WORKING-SET ALGORITHM



**Figure 10.22** Working-set model.



# PFF(PAGE-FAULT FREQUENCY) SCHEME

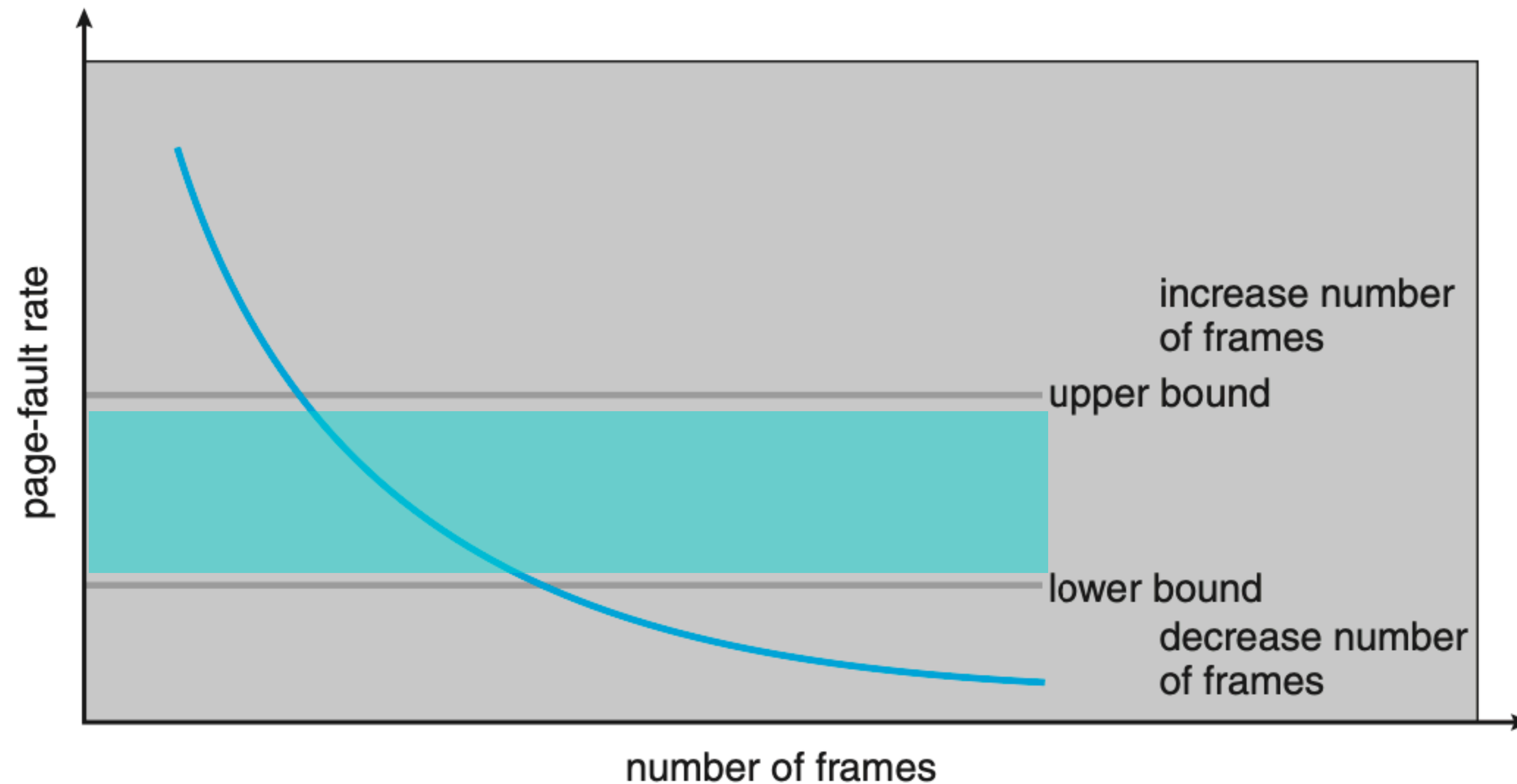


Figure 10.23 Page-fault frequency.

Page fault rate의 상한값과 하한값을 둔다.

Page fault rate이 상한값을 넘으면 frame을 더 할당  
하한값 이하이면 할당 frame 수를 줄임

빈 frame이 없으면 일부 프로세스를 swap out

# PAGE SIZE의 결정

---

Trend - Larger page size

Page size를 감소시키면?

- 페이지 수 증가
- 페이지 테이블 크기 증가
- 내부 조각 감소 (메모리를 조금 더 효율적으로 사용 할 수 있음)
- Disk transfer의 효율성의 감소
- 필요한 정보만 메모리에 올라와 메모리 이용엔 효율
- locality 활용 측면에서는 좋지 않음.

**감사합니다**

