OPERATING SYSTEMS

by
Marwa Yusuf

Lecture 15
Thursday 25-12-2020

Chapter 3 (3.4 to 3.4.6)
Memory Management

Page Replacement Algorithms

- At page fault, a page is chosen to be evicted to make room for the incoming one.
- If the page was modified, it must be written back to disk. Otherwise (like in code) , just overwrite.
- It’s wise to choose the least used page, otherwise, it will be needed again shortly, causing another fault, hurting performance.
- Just like cache, but with longer times.
- Q: the evicted page should belong to the same faulting process (fixing size available to a process)? Or not?

The Optimal Page Replacement Algorithm

- Easy to describe, impossible to implement.
- Choose the page that will be used further from now.
- The problem is: ….
- Using profiling, 1st run info can guide 2nd run.
  - These info are specific to this program with this input only.
- Used for comparison with other algorithms.

The Not Recently Used **(NRU)** Algorithm

- Using **R** and **M** bits.
  - They are updated at each reference, must be done by hardware.
  - Once set, only OS can reset it.
- Can be simulated by OS
  - Initially, all pages marked **absent**. When referenced, a page fault occurs, the page is brought, **R** is set, marked **read only**.
  - If it is to be written, another page fault, and set **M**, change mode to **read/write**.

NRU cont.

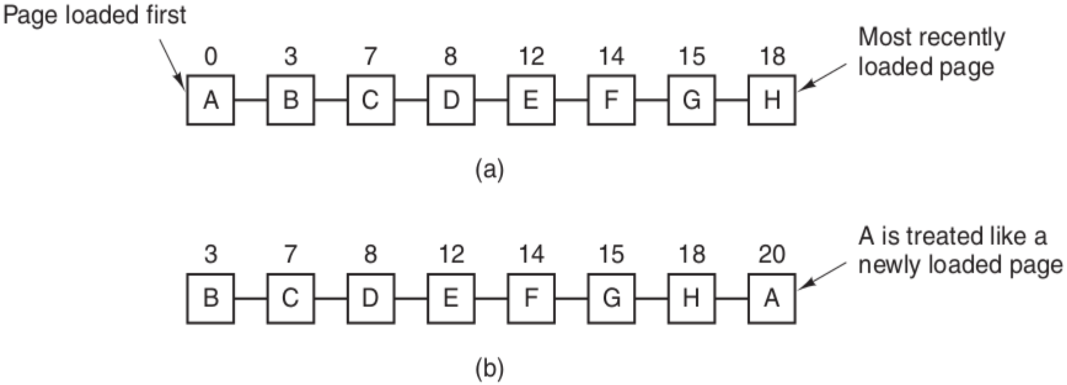
- Periodically (e.g. at clock interrupts) OS clears **R** bits.
- 4 possibilities:
  - Class 0: not referenced, not modified.
  - Class 1: not referenced, modified.
  - Class 2: referenced, not modified.
  - Class 3: referenced, modified.
- At replacement, choose from lowest-numbered nonempty class.
- Easy to understand, moderately efficient to implement and adequate performance.

FIFO Algorithm

- Keep a list of pages, with the oldest at head, which is removed when the need arises.
- However, the oldest may be important!
- SO, it is rarely used.

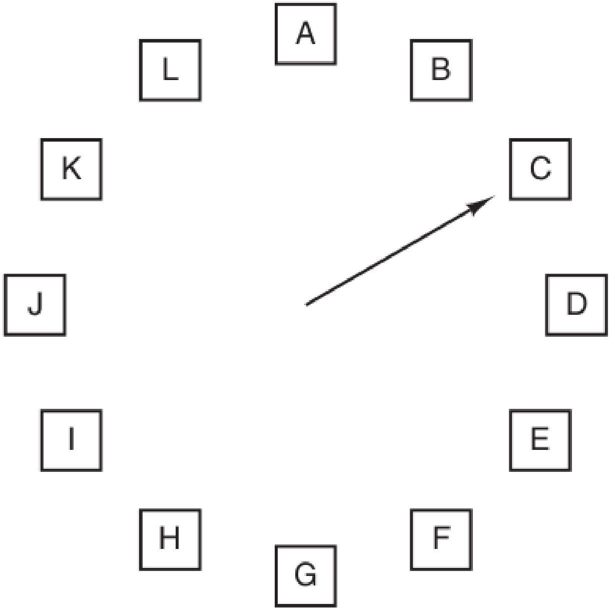
Second-Chance Algorithm

- A modification to FIFO
- Use **R** bit of the oldest page.
  - If **R = 0**, replace page.
  - If **R = 1**, reset it, move page to the tail (as if freshly loaded now.), and continue search with the next.
  - If all pages are referenced lately, it turns back to FIFO.



Clock Page Algorithm

- The same as second chance, but avoiding constantly moving pages in the list.



When a page fault occurs, the page the hand is pointing to is inspected. The action taken depends on the R bit:

R = 0: Evict the page
R = 1: Clear R and advance hand

Least Recently Used (**LRU**) Algorithm

- The page that is not used for a long time, most probably will remain unused. (approximation of optimal algorithm).
- It is expensive:
  - Should keep a list of pages, updated at each reference.
- Another alternative:
  - Keep a 64-bit counter (**C**) in hardware, incremented at each instruction.
  - Store **C** in the referenced page entry.
  - Replace page with the least **C**.