Memory Management – 4
Page Replacement Algorithms

CS 342 – Operating Systems
Ibrahim Korpeoglu
Bilkent University
Department of Computer Engineering

Page Replacement Problem

- When a page fault occurs, OS should choose a page to remove from memory to make room for the page that has to be brought in.
- If the page to be removed is modified, write it back to the hard-disk.
- A page to be evicted can be selected randomly, or by using an algorithm
  - It is better to select a page that is not heavily used.

Page Replacement Problem

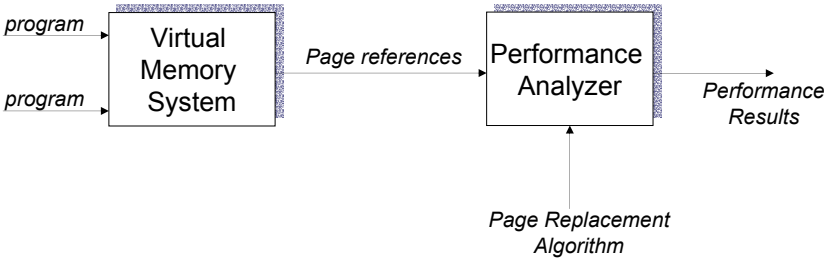
- Occurs in virtual memory systems and also in other areas of computer design.
  - Cache / Main Memory
    - Cache blocks need to be replaced
    - Every few nanoseconds
  - Web server
    - Web pages in memory need to be replaced
    - A web page is not modified – read only
  - Buffer / Database Management Systems

The Optimal Page Replacement Algorithm

- Assume a program is running.
  - At any moment, there are some set of pages belonging to that program
- Assume a page fault occurs.
  - Choose the page that will be used in far most point in time in the future.
    - Label each page with the number of instructions that must be executed before that page is first referenced.
    - Select the page that has the highest label to remove from main memory.

The Optimal Page Replacement Algorithm

- Easy to describe, but difficult to realize.
  - How can we in advance how many instructions will be executed before a page is first referenced?
- An approximation
  - Run the program on a simulator, keep track of all page references.
  - Implement the *optimal page replacement* algorithm on the second by using the trace you obtained in the first run.
- In this way, it is possible to compare other page replacement algorithms with the optimal one and decide how good they.



Other page replacement Algorithms

- The Not Recently Used Page Replacement Algorithm
- The First-In, First-Out Page (FIFO) Replacement Algorithm
- The Second Chance Page Replacement Algorithm
- The Clock Page Replacement Algorithm
- The Least Recently Used (LRU) Page Replacement Algorithm
- The Working Set Page Replacement Algorithm
- The WSClock Page Replacement Algorithm

The Not Recently Used Page Replacement Algorithm

- Associated with each page in memory, we have two bits:
  - *Referenced (R bit)*: Says if a page is referenced in the last clock interval
  - *Modified (M bit)*: Says if a page is modified after it has been loaded from disk to memory.
- These bits must be set with every memory access
  - Therefore, setting these bits must be very fast: usually in hardware.
- R bits are cleared with every clock interrupt

The Not Recently Used Page Replacement Algorithm

- Divide all page in memory into four classes
  - Class 0: not referenced, not modified
  - Class 1: not referenced, modified
  - Class 2: referenced, not modified
  - Class 3: referenced, modified
- Class 1 pages occur when a class 3 page (R bit and M bit set) has its R bit cleared.
- A modified but not referenced page has more chance for removal that a referenced but but modified page.
  - A clean page that is in heavy use should stay instead of a dirty page that is not recently used.

The Not Recently Used Page Replacement Algorithm

- **Algorithm**
  - When necessary, remove *randomly* a page from the *lowest* numbered class.
- Features
  - Simple to understand
  - Efficient to implement
  - Gives adequate performance.

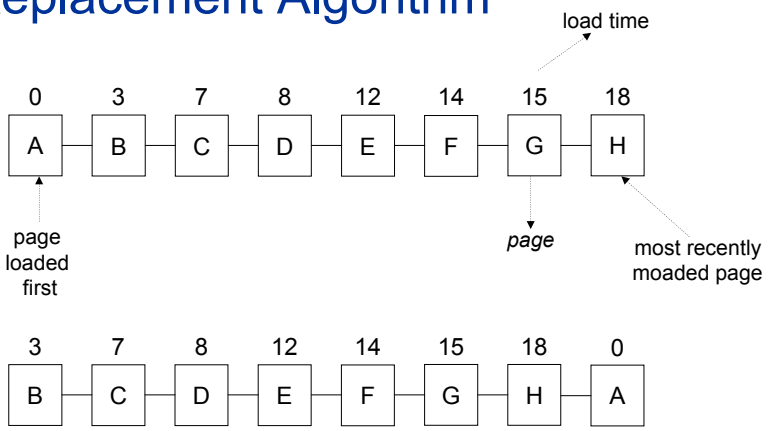
The FIFO Page Replacement Algorithm

- Pages in memory are maintained in a *linked list (a queue).*
  - *The page at the head is the oldest one.*
  - *The page at the tail is the newest one.*
- **Algorithm**:
  - When necessary, evict the page at the head of the list.
    - Add the new page to the tail of the list.
- Rarely used in its pure form.

The Second Chance Page Replacement Algorithm

- When it decides to remove a page
  - 1. Looks to the head of the list
  - 2. If head has R-bit as 1, then R-bit of head is cleared and it is put to the end.
  - Repeats steps 1 and 2 until the head contains a page which has R-bit zero.
- If R-bits of all pages in memory are set, then this algorithm is effectively same with FIFO.

The Second Chance Page Replacement Algorithm



Evict page 3!

The Clock Page Replacement Algorithm

- Keep all page frames on a circular list in the form of a clock.
- A hand points to the oldest page.
- If a page fault occurs:
  - 1. If R bit of page pointed to by hand is 0, then
    - remove the page;
    - Insert the new page to the same location,
    - Advance the hand by one position in clockwise direction.
  - 2. If R bit is 1, then hand is advanced by one position in clockwise direction, and R bit is set to zero.
    - Repeat step 2 until a page whose R-bit is 0 is found. Then execute step 1.

The Clock Page Replacement Algorithm