Memory Management

Week 08 – Tutorial

Page Replacement Algorithms &

Design Issues

Problem 3.27 ?

Suppose that the virtual page reference stream contains repetitions of long sequences of page references occasionally followed by a random page reference. For example, the sequence: 0, 1, ... , 511, 431, 0, 1, ... , 511, 332, 0, 1, ... consists of repetitions of the sequence 0, 1, ... , 511 followed by a random reference to pages 431 and 332

A. Why will the standard replacement algorithms (LRU, FIFO, clock) not be effective in handling this workload for a page allocation that is less than the sequence length?
B. If this program allocated 500 page frames, describe a page replacement approach that would perform much better than the LRU, FIFO, or clock algorithms

Problem 3.27 – Solution (1/2)

A. Consider, for example, a page allocation scheme with 510 frames. First 510 references will generate page faults because none of the pages are in memory. The next reference will generate page fault too and also will push page 1 out of memory. Therefore, every reference will page fault unless the number of page frames is 512, the length of the entire sequence.

Problem 3.27 – Solution (2/2)

B. If there are only 500 frames, the alternative approach would be to map pages 0–498 to fixed frames and vary only one frame

Problem 3.28

- If FIFO page replacement is used with four page frames and eight pages, how many page faults will occur with the reference string 0172327103 if the four frames are initially empty?
- Now repeat this problem for LRU

Problem 3.28 – Solution (1/2)

- The page frames for FIFO are as follows:

Page frames

x | 0 | 1 | 7 | 2 | 3 | 3 | 3 | 3 | 0 | 0
x | x | 0 | 1 | 7 | 2 | 2 | 2 | 2 | 3 | 3
x | x | x | 0 | 1 | 7 | 7 | 7 | 7 | 2 | 2
x | x | x | x | 0 | 1 | 1 | 1 | 1 | 7 | 7

Reference string: 0172327103

Problem 3.28 – Solution (2/2)

- The page frames for LRU are as follows:

x | 0 | 1 | 7 | 2 | 3 | 2 | 7 | 1 | 0 | 3

x | x | 0 | 1 | 7 | 2 | 3 | 2 | 7 | 1 | 0

x | x | x | 0 | 1 | 7 | 7 | 3 | 2 | 7 | 1

x | x | x | x | 0 | 1 | 1 | 1 | 3 | 2 | 7

Reference string: 0172327103

Updates only the order in the linked list

Problem 3.30

- A small computer on a smart card has four page frames. At the first clock tick, the R bits are 0111 (page 0 is 0, the rest are 1). At subsequent clock ticks, the values are 1011, 1010, 1101, 0010, 1010, 1100, and 0001. If the aging algorithm is used with an 8-bit counter, give the values of the four counters after the last tick

Problem 3.30 – Solution (1/3)

- The counters after the first tick are:

Page 0: 0 0 0 0 0 0 0 0

Page 1: 1 0 0 0 0 0 0 0

Page 2: 1 0 0 0 0 0 0 0

Page 3: 1 0 0 0 0 0 0 0

R bits

See "3.4.7 Simulating LRU in Software" (TB page 214)

Problem 3.30 – Solution (2/3)

- The counters after the second tick are:

Page 0: 1 0 0 0 0 0 0 0

Page 1: 0 1 0 0 0 0 0 0

Page 2: 1 1 0 0 0 0 0 0

Page 3: 1 1 0 0 0 0 0 0

Problem 3.30 – Solution (3/3)

- The final counters are:

  Page 0: 0 1 1 0 1 1 1 0 $= 110_{10}$

  Page 1: 0 1 0 0 1 0 0 1 $= 73_{10}$

  Page 2: 0 0 1 1 0 1 1 1 $= 55_{10}$

  Page 3: 1 0 0 0 1 0 1 1 $= 139_{10}$

Problem 3.31 ?

For which page reference sequence the first page selected for replacement will be different for the clock and LRU page replacement algorithms? Assume that a process allocated 3 frames, and the reference string contains page numbers from 0 to 3.

A. 0 1 2 3

B. 0 2 0 0 0 1 1 2 3

C. 0 1 2 1 2 0 3

Problem 3.31 – Solution (1/3)

- Clock algorithm:
  - 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
- LRU:
  - When a page fault occurs, throw out the page that has been unused for the longest time

Problem 3.31 – Solution (2/3)

- Let's consider a simple reference string 0123:
  - According to clock algorithm, pages will be organized as a circle and the hand will point to page 0.
    If we refer page 3, then R bits of every page will be cleared and on the second round page 0 will be evicted from memory
  - According to LRU, page 0 will be evicted too since it has been unused for the longest time

Problem 3.31 – Solution (3/3)

- Consider the sequence 0 1 2 1 2 0 3
  - In LRU, page 1 will be replaced by page 3
  - In clock, page 0 will be replaced, since all pages will be marked (R=1) and the cursor is at page 0

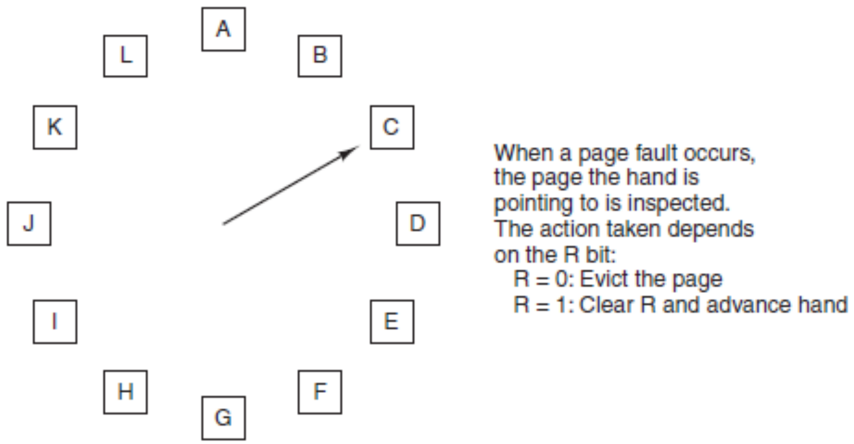
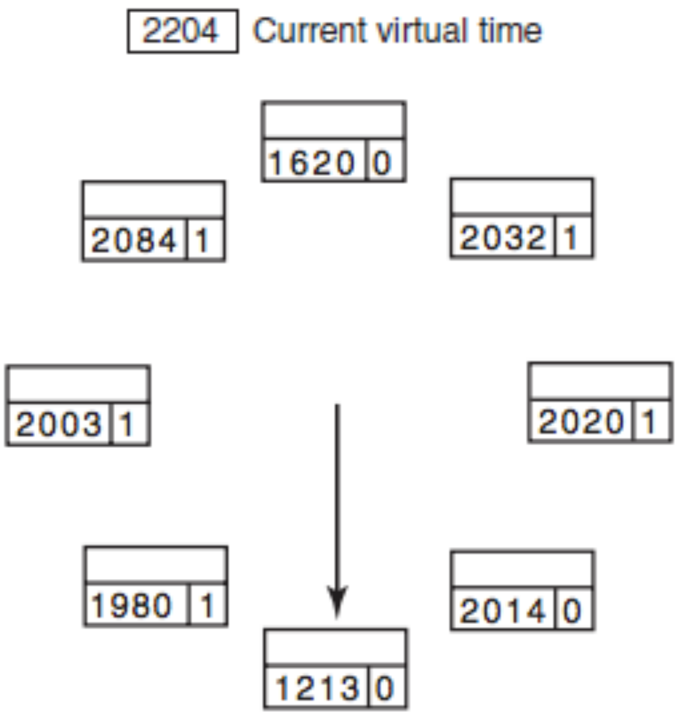


Figure 3-16. The clock page replacement algorithm.

Extracted from TB page 213

Problem 3.32

- In the WSClock algorithm of Fig. 3-20(c), the hand points to a page with
  R = 0
  If τ = 400, will this page be removed?
- What about if τ = 1000?



Problem 3.32 – Solution

- The age of the page is 2204 − 1213 = 991
- If $\tau = 400$, it is definitely out of the working set and was not recently referenced so it will be evicted
- The $\tau = 1000$ situation is different. Now the page falls within the working set (barely), so it is not removed

Problem 3.33 (1/2)

- Suppose that the WSClock page replacement algorithm uses a τ of two ticks, and the system state is the following (V, R, and M stand for Valid, Referenced, and Modified):

Valid: indicates whether the entry is in use or not (See TB pg 202)

Page	Time stamp	V	R	M
0	6	1	0	1
1	9	1	1	0
2	9	1	1	1
3	7	1	0	0
4	4	0	0	0

Problem 3.33 (2/2)

A. If a clock interrupt occurs at tick 10, show the contents of the new table entries. Explain. (You can omit entries that are unchanged.)

B. Suppose that instead of a clock interrupt, a page fault occurs at tick 10 due to a read request to page 4. Show the contents of the new table entries. Explain. (You can omit entries that are unchanged.)

Problem 3.33 – Solution (1/2)

- Consider,
  A. For every R bit that is set, set the time-stamp value to 10 and clear all R bits. You could also change the (0,1) R-M entries to (0,0*). So the entries for pages 1 and 2 will change to:

See TB page 219

Page	Time stamp	V	R	M
0	6	1	0	0*
1	10	1	0	0
2	10	1	0	1

Problem 3.33 – Solution (2/2)

B. Evict page 3 (R = 0 and M = 0) and load page 4:

Page	Time stamp	V	R	M	Notes
0	6	1	0	1	
1	9	1	1	0	
2	9	1	1	1	
3	7	0	0	0	Changed from 7 (1,0,0)
4	10	1	1	0	Changed from 4 (0,0,0)

End

Week 08 – Tutorial