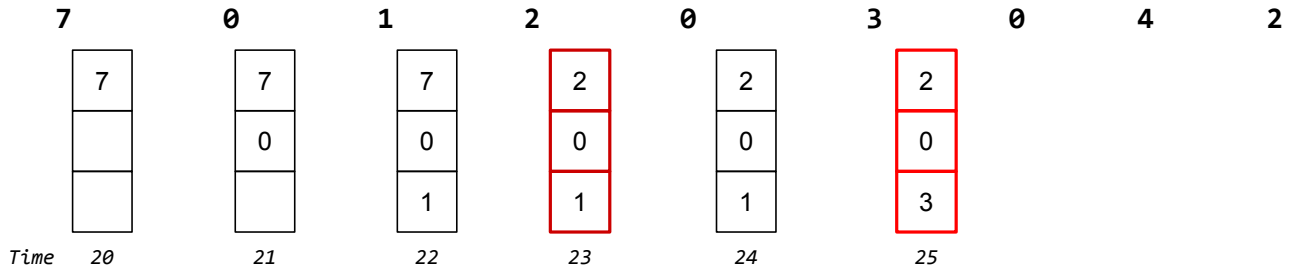
LRU Implementation (HW / SW cooperation)

LRU Implementation (General Idea)



Victim selection criteria:

- *Theoretical*: **max** of {backward "*distance*" / previous ref}
- *Practical*: **min** of {"*time*" to the *most recent* reference}

LRU Timestamp

Choice of implementation

- Absolute timestamp: CPU clock or counter
  - Require additional memory write to the page table (by hardware) on **every memory reference**
- Relative timestamp: linked list of page numbers (*hard to do in hardware*)
  - Associated with each process is a list of page numbers
  - When a page is referenced, move the corresponding page number to the **head** end
  - Victime page is the page at the **tail** end of the list

LRU Approximations

- Pure implementation of LRU requires accurate timestamp
  - Updating an N-bit timestamp is costlier than updating a 1-bit data
- Approximation Techniques (approximate timestamp with tags)
  - Bit in tags are set to one by HW, reset to zero by SW
  - k-bit history tag
  - Second-chance algorithm
  - Enhanced second-chance algorithm

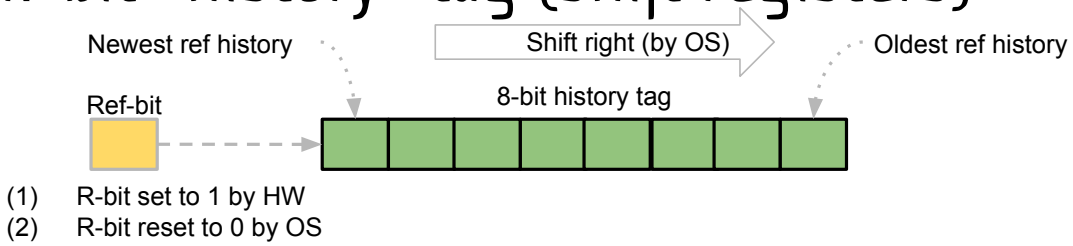
Approximation #1: reference-bit

- Use a reference-bit in place of actual (absolute) timestamp
  - Include the ref-bit on each cell/row of page table (in addition to the *modified-bit*)
- Updating the ref-bit
  - Set to one by **hardware** on every memory reference
  - Reset to zero by **software** (the OS) when the page is swapped in
- Victim Selection: page whose ref-bit is zero
  - Cons: ***can't distinguish reference 5ms (larger timestamp) vs 20ms (smaller timestamp) ago***

Approximation #2: *k-bit "timestamp"*

- Use a **k-bit history tag** (in addition to the ref-bit) for better approximation of timestamp
  - **Hardware**: the 1-bit ref is automatically set to ONE **by MCU**
  - **Software**: the OS *periodically*:
    - shifts right the k-bit history and inject the ref-bit at the MSB position of the k-bit history
    - Reset the ref-bit
- Victim Selection: page whose k-bit history is lowest (treated as unsigned integer)
  - Use FIFO to break tie

k-bit "history" tag (shift registers)



(1) R-bit set to 1 by HW
(2) R-bit reset to 0 by OS

Assuming periodic update by OS every 100ms

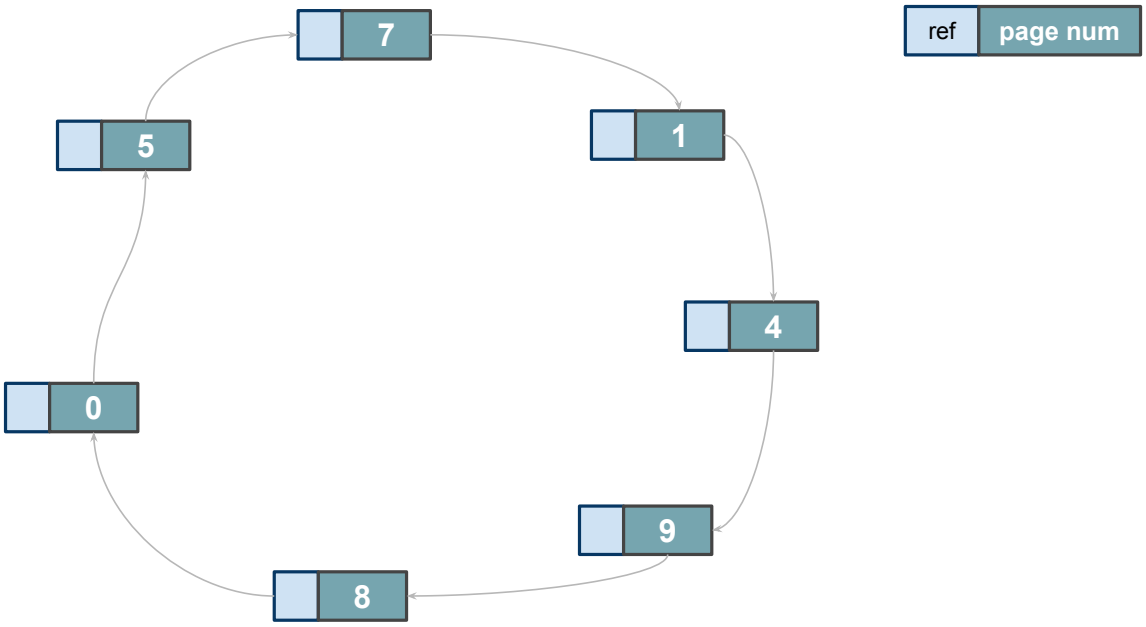
- `00000000`: never referenced within the last 800 ms
- `01100001`: referenced within the last 200ms, 300ms, and 800ms **ago**
- `00010010`: referenced within the last 400ms and 700ms **ago**

Relative timestamp

Approximation #3: 1-bit Clock Algorithm

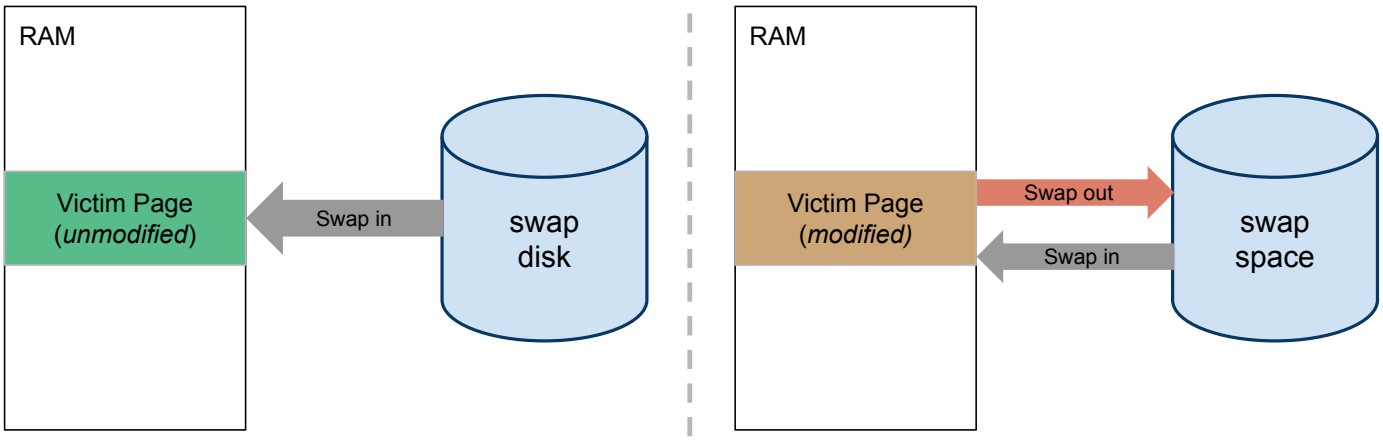
- Based on FIFO (uses a circular buffer to maintain the resident page numbers)
- Algorithm
  - **Hardware**: set Ref-bit to ONE on each memory reference
  - **Software**
    - Advance the buffer pointer to locate a page whose ref-bit is ZERO
    - *While advancing the pointer*, ***reset non-zero reference-bit to ZERO***
    - The next search begins at the next position where the last victim was found
- The worst case: all the ref-bits were ONE, the algorithm would have inspected all the pages before finding one victim, giving all the pages a **second chance** (to stay resident)

1-bit Second Chance (Clock Algorithm)



ref-bit is automatically set to 1 by **hardware** when a page is referenced, reset to 0 by **software** during search

Swap In and Swap Out



I/O cost: 1 disk read

I/O cost: 1 disk read + 1 disk write

Approximation #4: 2-bit Clock Algorithm

- Improvement to 1-bit clock algorithm, but uses the combination of both ref-bit and modified-bit
- Four types of page

Class	ref-bit	mod-bit	Description
0	0	0	Not (recently)-used and not modified
1	0	1	Not (recently)-used but modified
2	1	0	(recently)-used and not modified
3	1	1	(recently)-used and modified

- Victim selection: select from the lowest class available

1-bit vs. 2-bit Clock Algorithms

R-bit **1-bit clock**

- Set to 1 by MCU (hardware)
- Reset to 0 by PFI handler (software)
  - At swap in
  - During search

R-bit **2-bit clock**

- Set to 1 by MCU (hardware)
- Reset to 0 by PFI handler (software)
  - At swap in
  - During search

M-bit

- Set to 0 by MCU on memory read
- Set to 1 by MCU on memory write
- Reset to 0 by PFI handler **only** at swap in
- Untouched by PFI handler during search

Two-Bit Clock Algorithm: Multi-Pass Search

- Look for Class 0: not-used, not-modified (R:0, M:0)
  - While searching, also reset Ref-bit to zero
    - Class 2 (**R=1**, M=0) becomes class 0 (**R=0**, M=0)
    - Class 3 (**R=1**, M=1) becomes class 1 (**R=0**, M=1)
- If no victim found, search again for "Class 0" (originally *were* class 2)
- If no victim found (yet), search again for Class 1 (include those originally *were* class 3)
- *Overall effect*: **modified pages have lower priority** to be kicked out

2-bit Second Chance (Enhanced Clock Algorithm)