

CIS 452 - Operating Systems Concepts

Nathan Bowman

Images taken from Silberschatz book

---

Approximating LRU (Second Chance)

Assume once again that OS is approximating LRU using hardware support for just single reference bit per page

Previously, OS "cheated" and kept track of additional information itself

We now examine algorithms that make use of just one bit

**Second-chance** algorithm is FIFO with a twist

When page fault requiring replacement occurs, OS examines pages in FIFO order

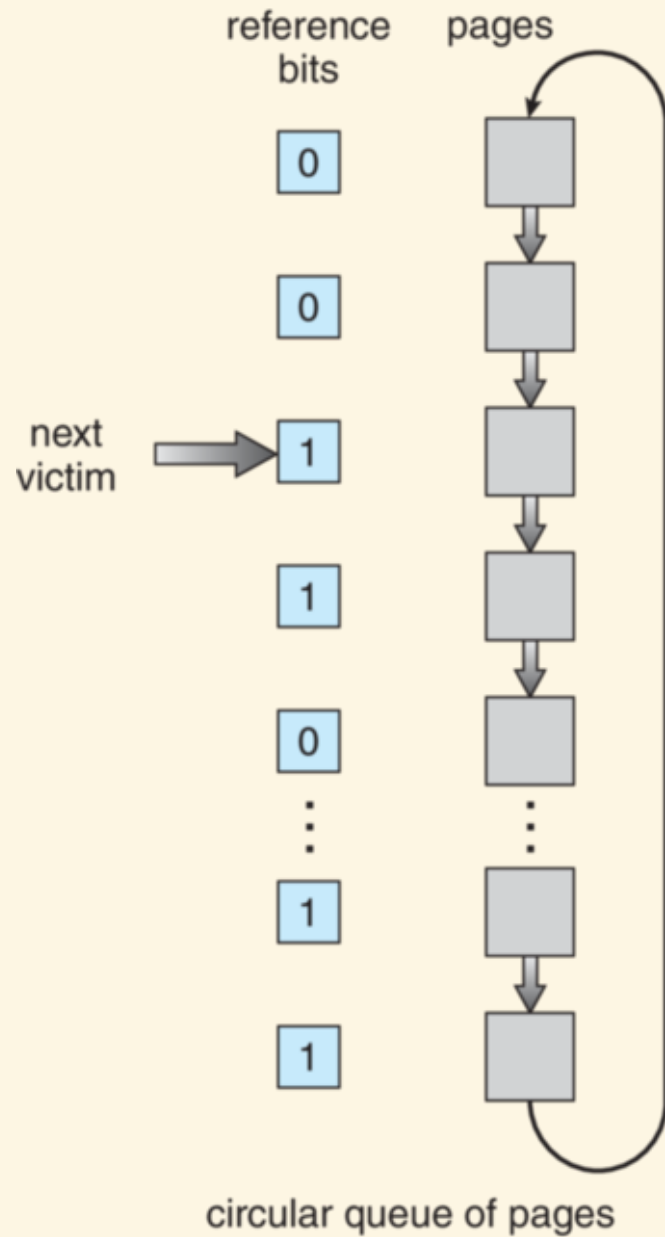
However, if reference bit of oldest page is set, it does not get removed and instead gets "second chance"

This keeps memory that is actually used from being evicted unless there is no other option

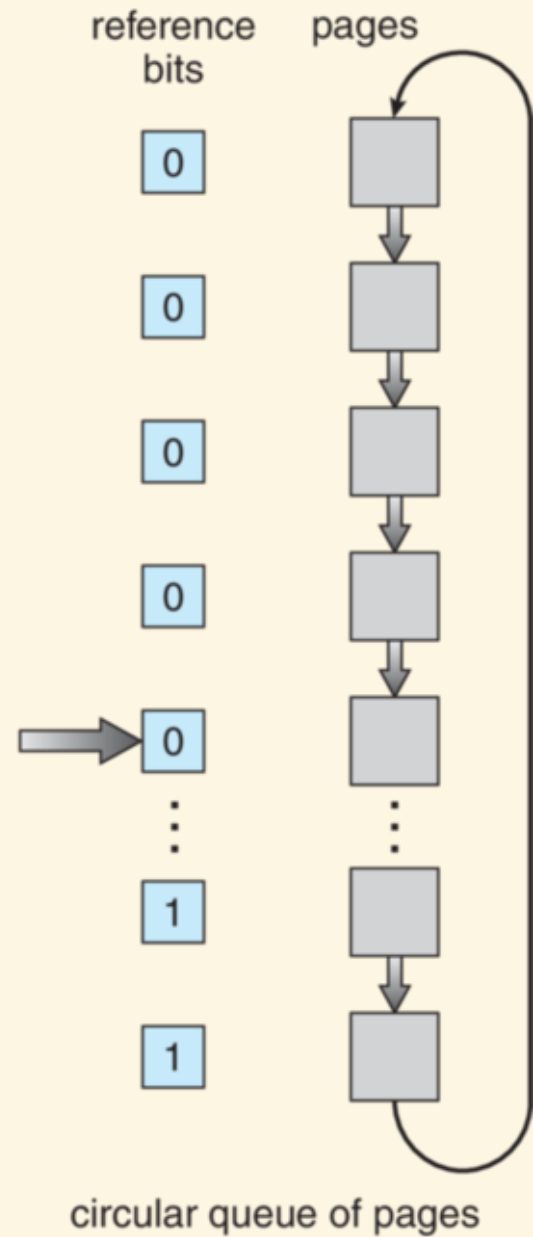
## Second chance

---

- Select oldest page
- If reference bit not set, select page as victim
- Otherwise
  - Unset reference bit
  - Move process to end of queue (most-recently used)
  - Repeat with updated oldest page



(a)



(b)



After page has reference bit checked and bit is active,  
page will not be removed until all other pages have  
either been removed or had their own reference bits  
checked

Page that is very active may have reference bit set again  
before it is tested again

Frequently used pages can stay in memory indefinitely

Note that if all pages had reference bit set when page fault occurred, second-chance algorithm degenerates to  
FIFO



In enhanced second-chance algorithm, both reference bit and modify bit are used

Kicking out modified pages is twice as expensive as kicking out unmodified ones

Algorithm gives preference to keeping modified pages, but only as tiebreaker after considering reference bit

Reference	Modify	Value
0	0	00
0	1	01
1	0	10
1	1	11

Proceeds in same way as second chance except two-bit value is used in place of single reference bit

OS searches list in FIFO order, replacing first page in lowest non-empty class

May take several iterations through queue before page selected for removal

Non-LRU algorithms are also studied

Counting-based algorithms keep track of number of page accesses rather than time of accesses

This also requires hardware support and turns out not to be efficient or accurate, so we will not consider this class of algorithms further