Page Replacement Algorithms

FIFO, NRU, LRU, NFU...

Index

Plan of Action

- What is paging?
- What is page replacement?
- What are the types of page replacement?
- Why we need a page replacement algorithm?
- What are the algorithms?

What is Paging?

- The OS divides virtual memory and the main memory into units, called pages.
- Each used page can be either in secondary memory or in a page frame in main memory.
- A frame does not have to comprise a single physically contiguous region in secondary storage.

What is page replacement?

- When memory located in secondary memory is needed, it can be retrieved back to main memory.
- Process of storing data from main memory to secondary memory ->**swapping out**
- Retrieving data back to main memory ->**swapping in**

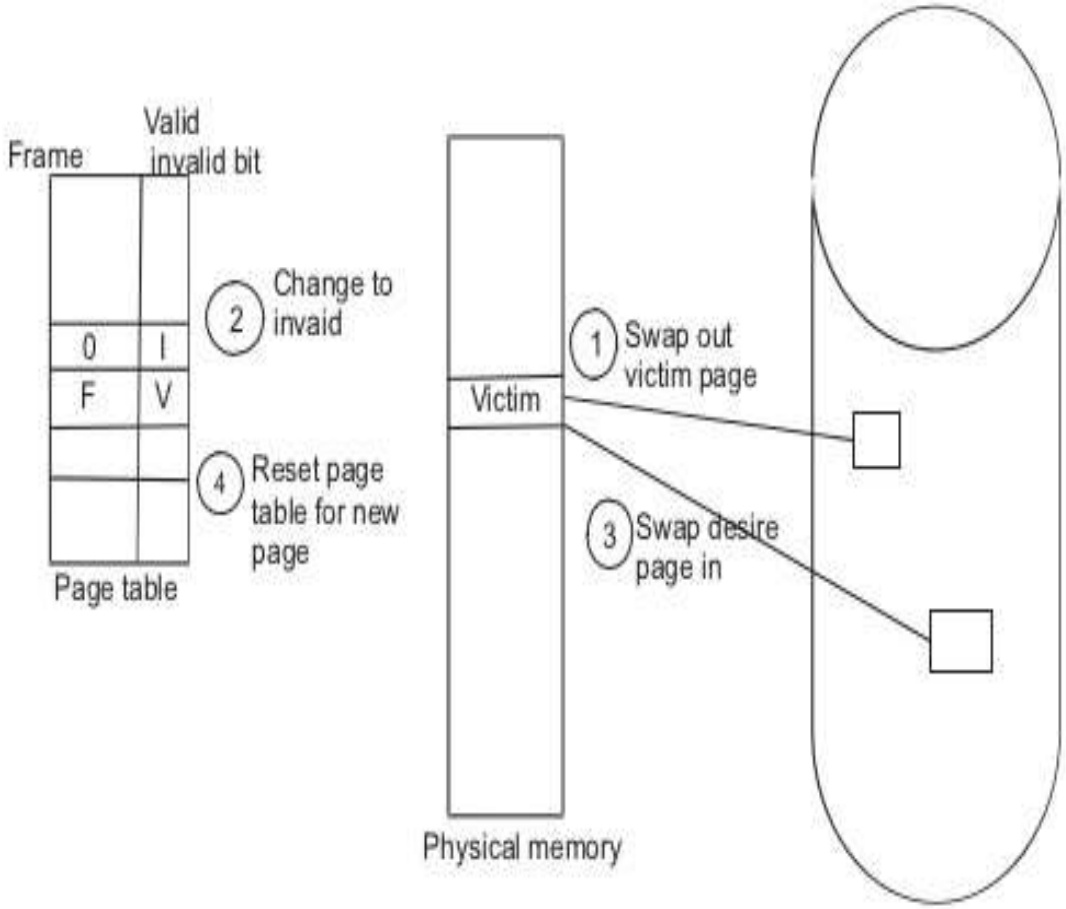


Fig: Page Replacement

What are Page Replacement Algorithms?

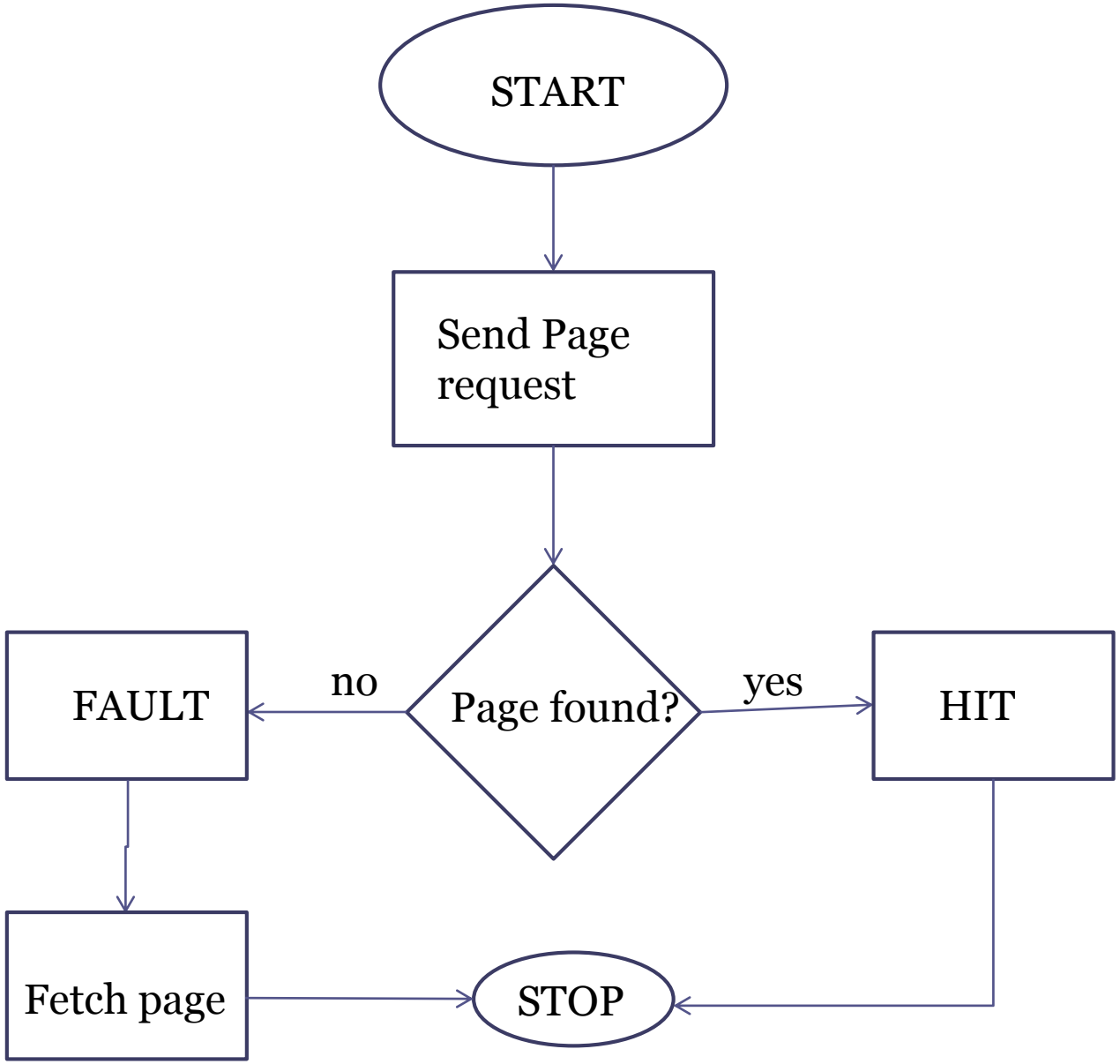
- Deals with which pages need to be swapped out and which are the ones that need to be swapped in
- The efficiency lies in the least time that is wasted for a page to be paged in

Types

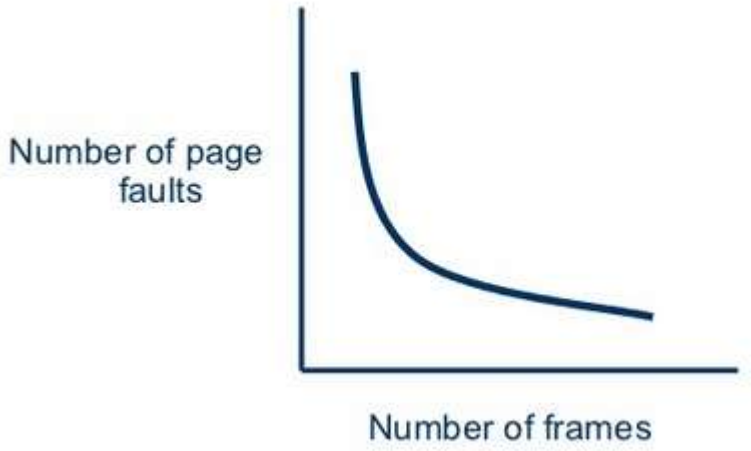
- Local Page Replacement Strategy
- Global Page Replacement Strategy

Why we need a page replacement algorithm?

- The main goal of page replacement algorithms is to provide lowest page fault rate.


START
Send Page request
Page found?
no
yes
FAULT
HIT
Fetch page
STOP

No. of Page Faults Vs No. of Frames



Algorithms

- First In First Out
- Optimal Replacement
- Not Recently Used
- Second Chance
- CLOCK
- Not Frequently Used
- Least Recently Used
- Random Replacement
- Working Set Replacement

First-In First-Out (FIFO)

- Pages in main memory are kept in a list
- Newest page is in head and the oldest in tail
- It does not take advantage of page access patterns or frequency

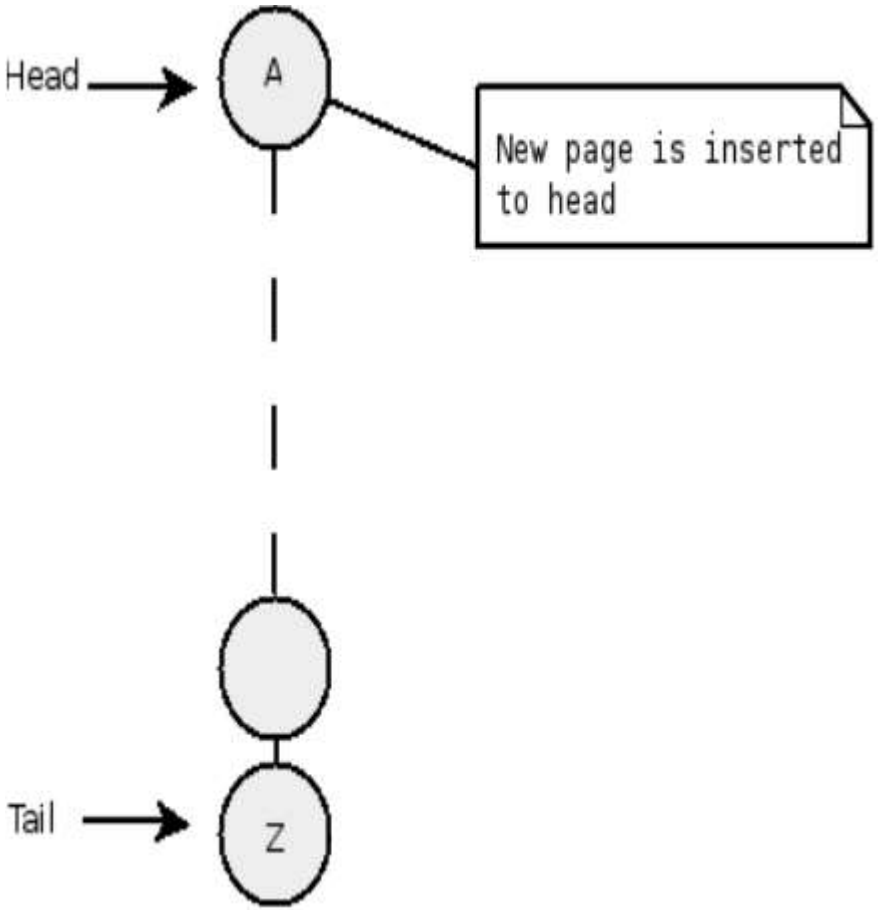


Fig: FIFO

FIFO Example

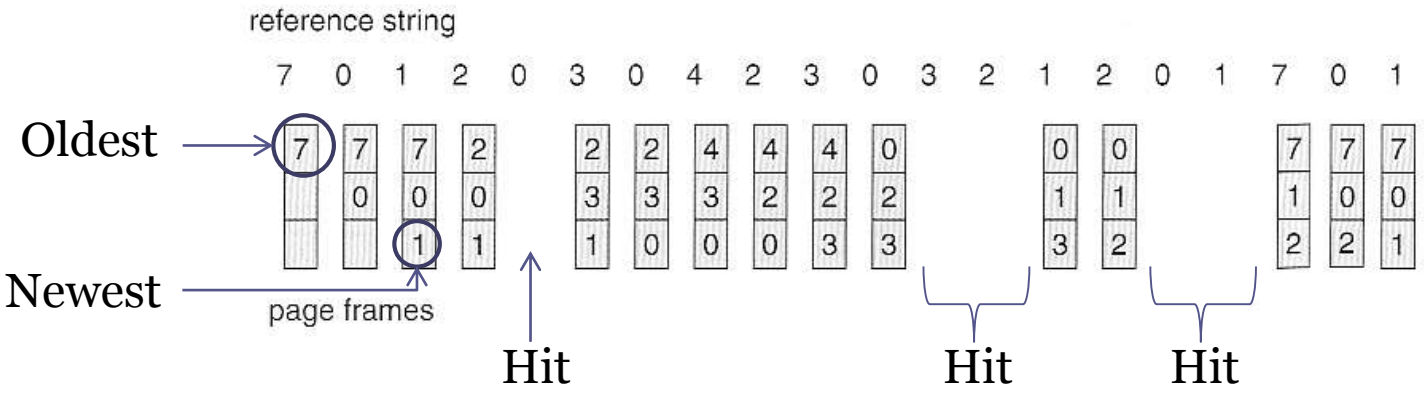


Fig: FIFO example

Optimal Replacement (OPT)

- When the memory is full, evict a page that will be unreferenced for the longest time
- The OS keeps track of all pages referenced by the program
- Only if the program's memory reference pattern is relatively consistent

OPTIMAL Example

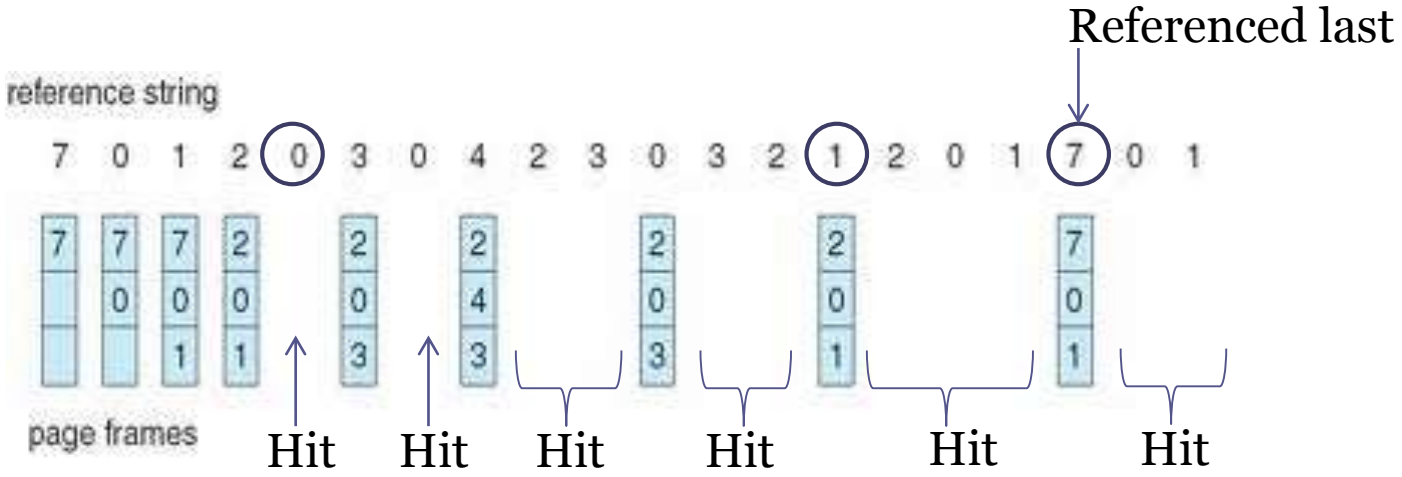


Fig: OPTIMAL example

Not Recently Used (NRU)

- It favours keeping pages in memory that have been recently used.
- The OS divides the pages into four classes based on usage during the last clock tick:
  - 3. Referenced, modified
  - 2. Referenced, not modified
  - 1. Not referenced, modified
  - 0. Not referenced, not modified

NRU

- Pick a random page from the lowest category for removal
- i.e. the not referenced, not modified page

NRU Example

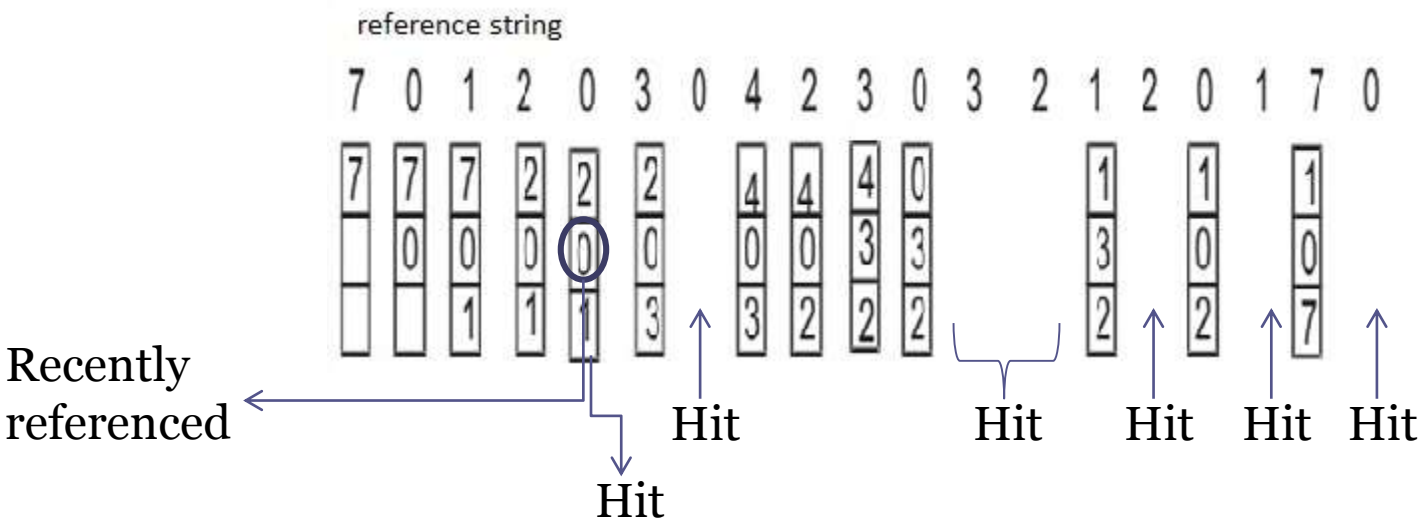


Fig: NRU example

Second Chance

- Modified version of FIFO
- Instead of swapping out the last page, the referenced bit is checked
- Gives every page a "second-chance"

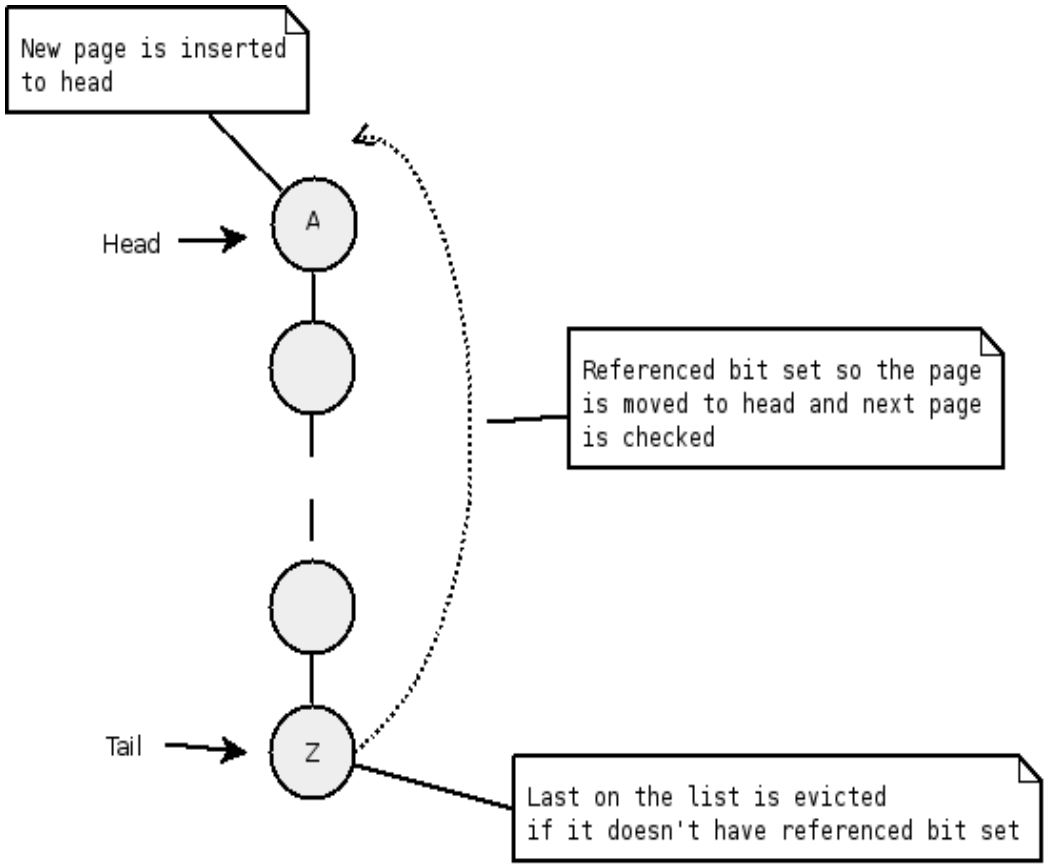


Fig: Second Chance

Clock

- Modified version of FIFO
- The set of frame candidates for replacement is considered as a circular buffer.

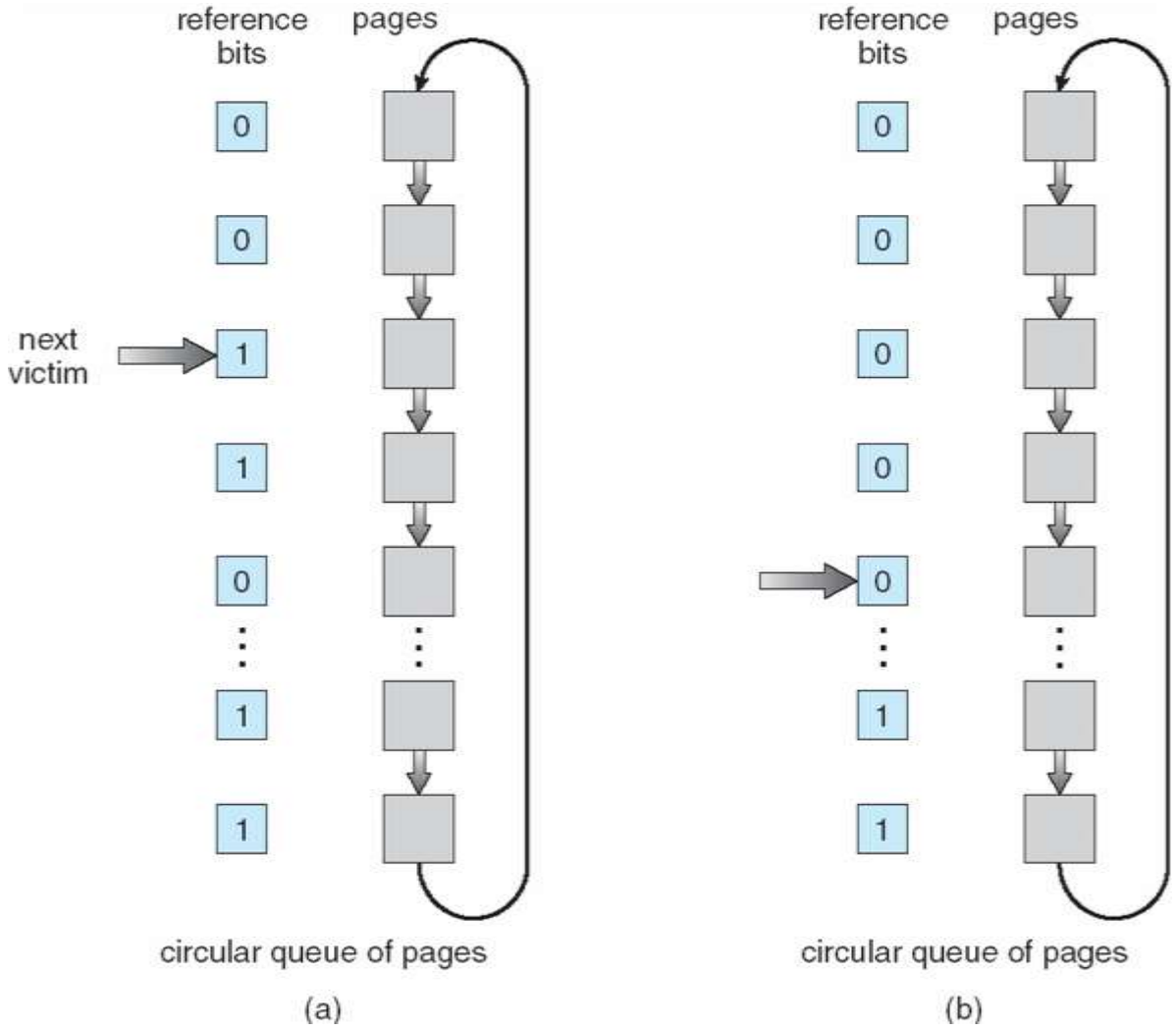

reference bits
pages
0
0
next victim
1
1
0
1
1
circular queue of pages
(a)
reference bits
pages
0
0
0
0
0
1
1
circular queue of pages
(b)

Fig: CLOCK

Least Recently Used (LRU)

- It swaps the pages that have been used the least over a period of time.
- It is free from Belady's anomaly.

LRU Example

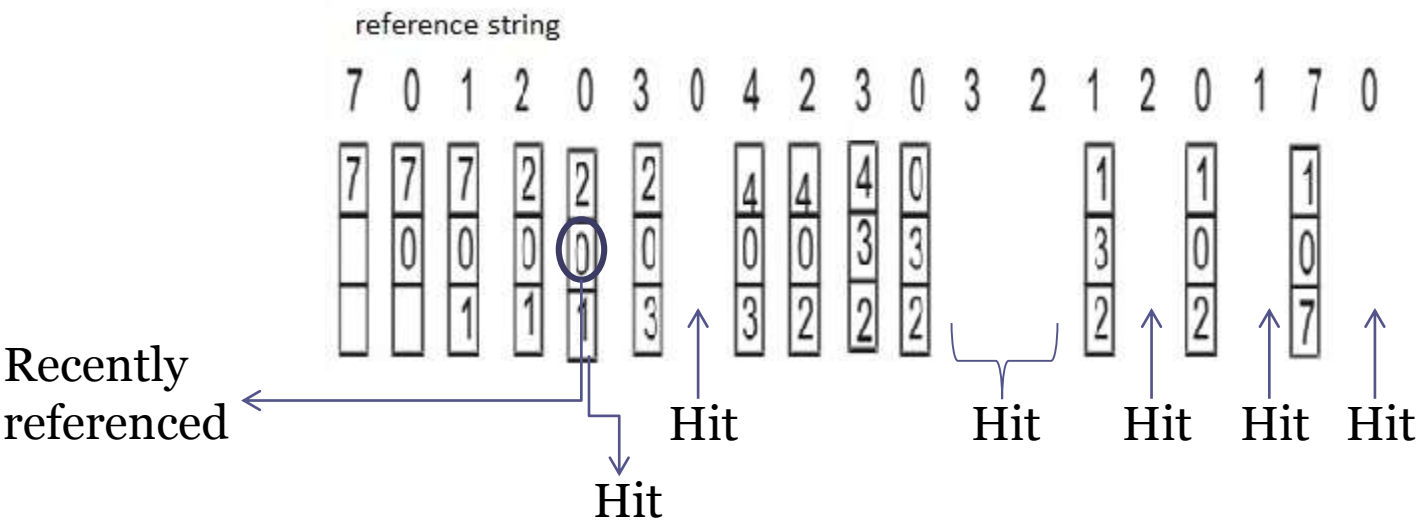
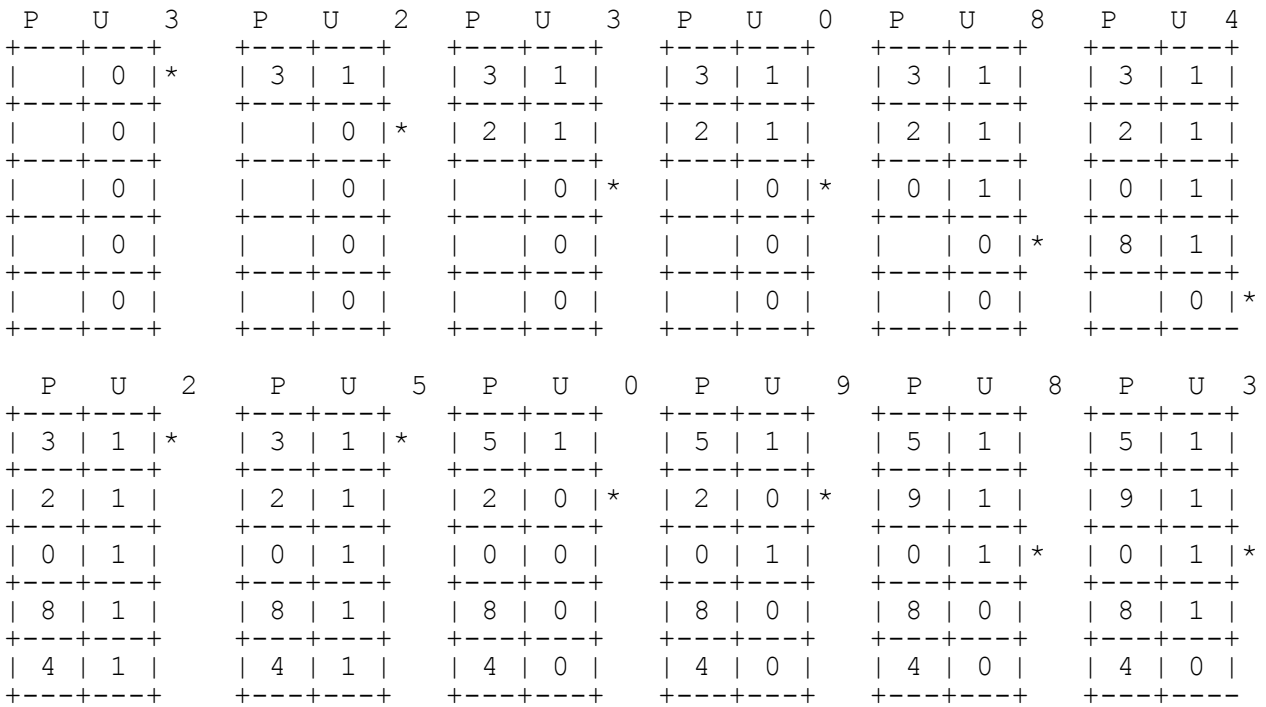


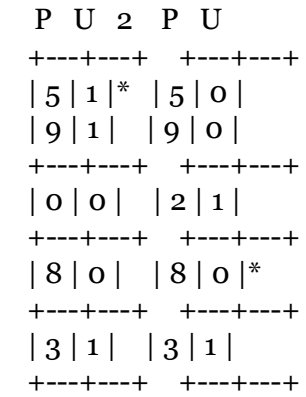
Fig: LRU example

Not frequently used (NFU)

- This page replacement algorithm requires a counter
- The counters keep track of how frequently a page has been used
- The page with the lowest counter can be swapped out

reference sequence : 3 2 3 0 8 4 2 5 0 9 8 3 2





* = indicates the pointer which identifies the next location
to scan P = page# stored in that frame U = used flag
0 = not used recently 1 = referenced recently

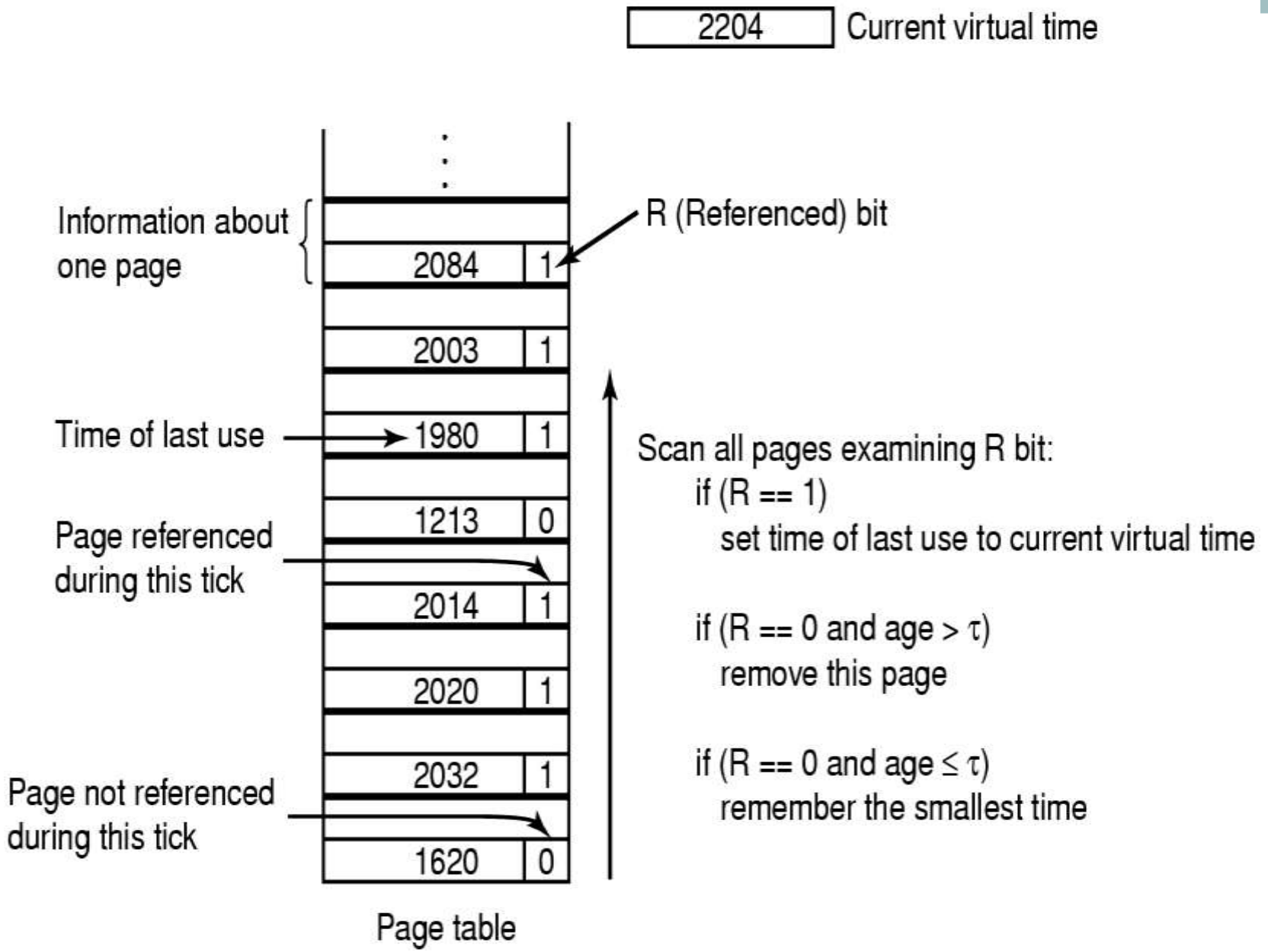
Fig: NFU example

Random

- This algorithm replaces a random page in memory.
- It fares better than FIFO.

Working set page replacement

- The set of pages that a process is currently using is called the working set.
- The working set algorithm is based on determining a working set and evicting any page that is not in the current working set upon a page fault.


2204
Current virtual time
Information about one page
R (Referenced) bit
2084
1
2003
1
Time of last use
1980
1
1213
0
Page referenced during this tick
2014
1
2020
1
2032
1
Page not referenced during this tick
1620
0
Page table
Scan all pages examining R bit:
if (R == 1)
set time of last use to current virtual time
if (R == 0 and age > τ)
remove this page
if (R == 0 and age ≤ τ)
remember the smallest time

Conclusion

Algorithm	Comment
• FIFO	• Might throw out important pages
• OPTIMAL	• Not implementable
• LRU	• Excellent but difficult to implement
• NRU	• Crude approximation of LRU
• NFU	• Crude approximation of LRU
• Second Chance	• Big improvement over FIFO
• CLOCK	• Realistic

References

- Web Links
  - www.wikipedia.com
  - www.youtube.com
  - www.vbForum.com
- Papers
  - Operating System Page Replacement Algorithms by A. Frank C. Wersberg
- Books
  - Computer Organization & Architecture by William Stallings

Thank You