Memory Management
Lecture 8

Page Replacement Algorithm

Minakshi R.

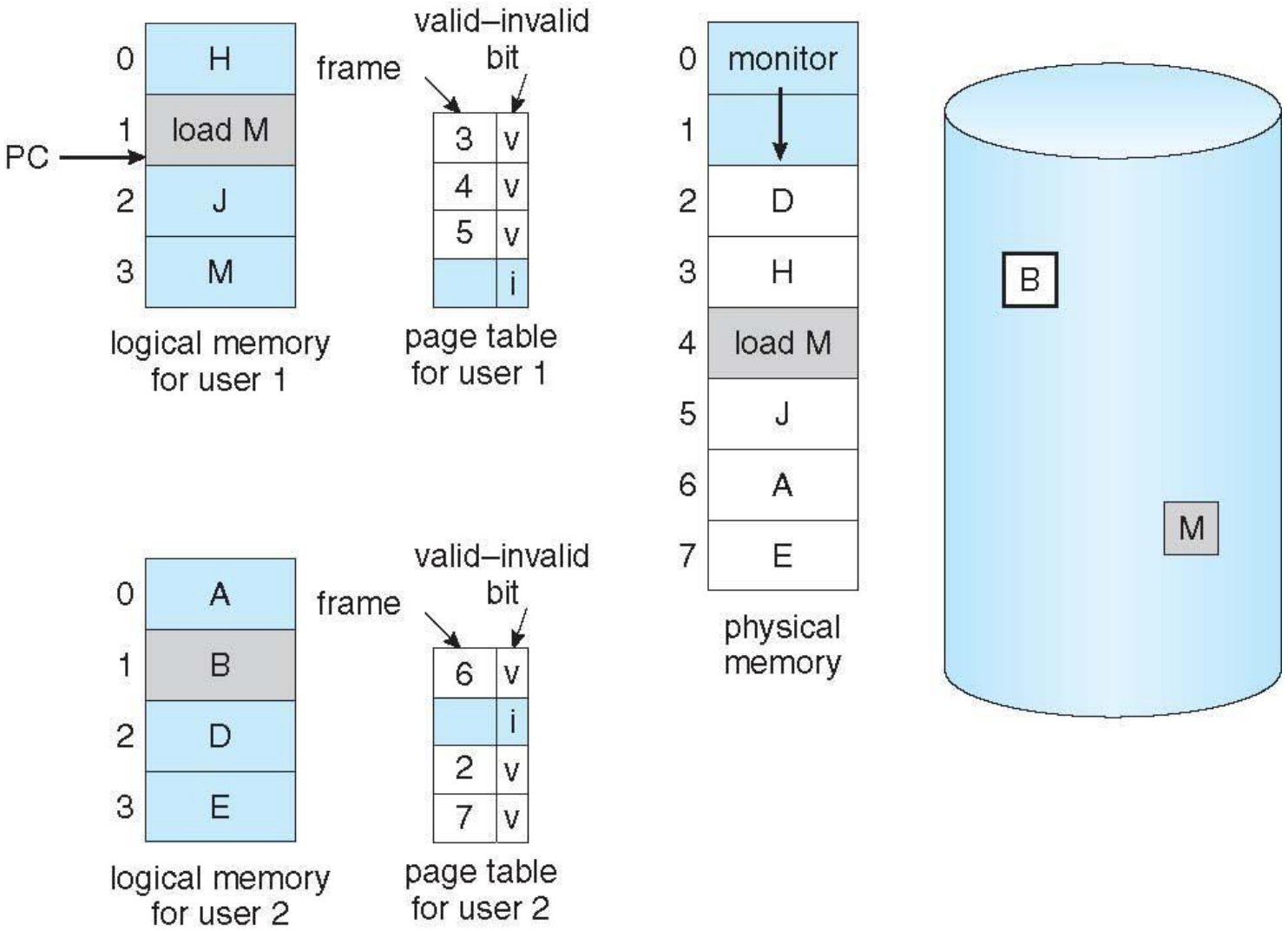
Operating System Concepts 8th edition silberschatz Galvin

Page Replacement

- Prevent over-allocation of memory by modifying page-fault service routine to include page replacement

- Page replacement completes separation between logical memory and physical memory – large virtual memory can be provided on a smaller physical memory

Need For Page Replacement

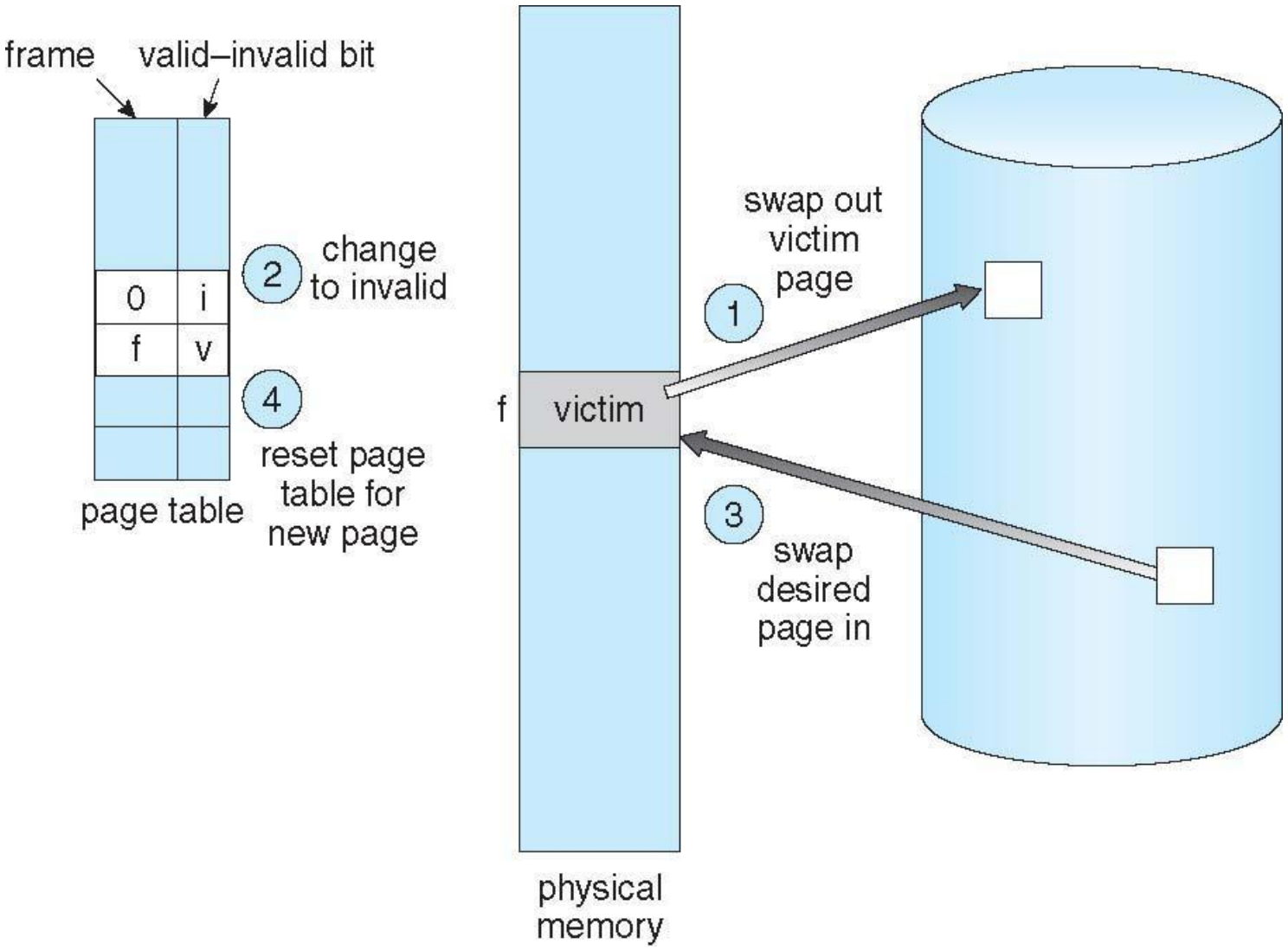


Basic Page Replacement

1. Find the location of the desired page on disk
2. Find a free frame:
   - If there is a free frame, use it
   - If there is no free frame, use a page replacement algorithm to select a **victim frame**
   - Write victim frame to disk

   Bring the desired page into the (newly) free frame; update the page and frame tables
3. Continue the process by restarting the instruction that caused the trap

Page Replacement

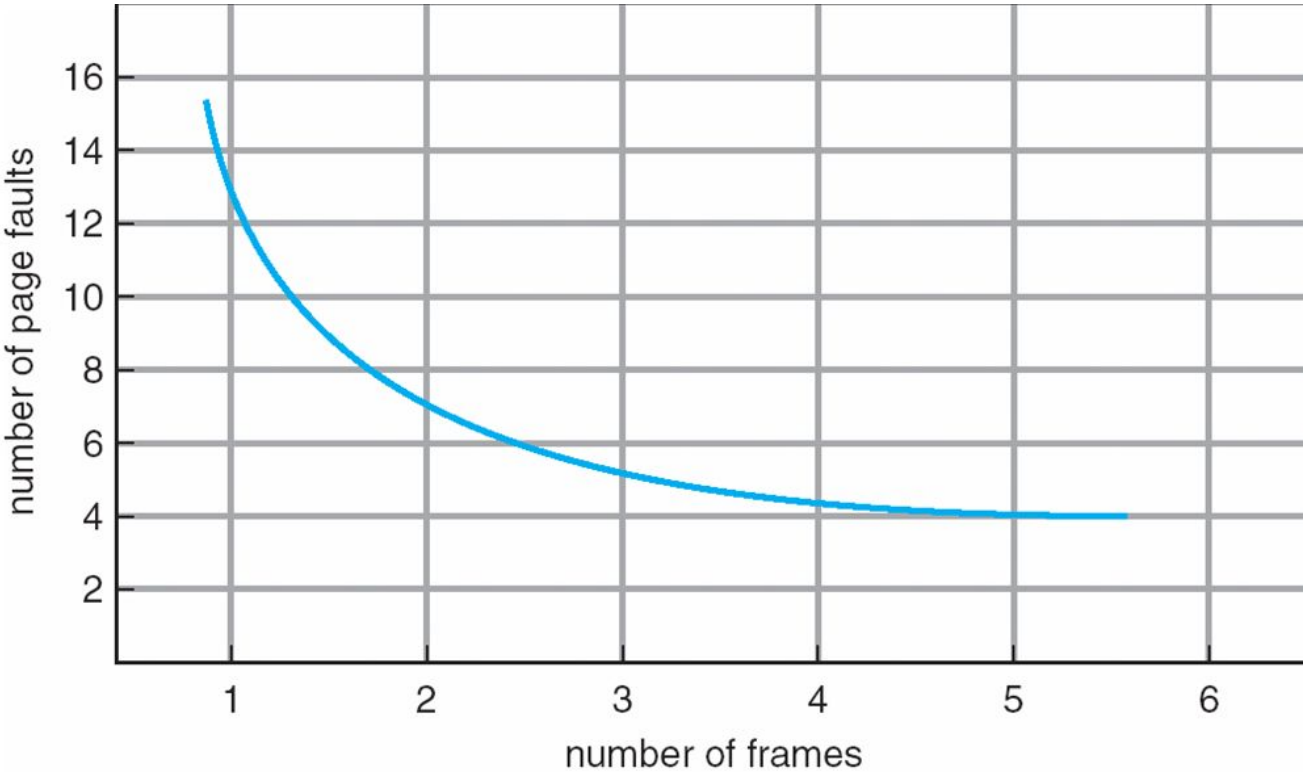


Page and Frame Replacement Algorithms

- **Frame-allocation algorithm** determines
  - How many frames to give each process
  - Which frames to replace
- **Page-replacement algorithm**
  - Want lowest page-fault rate on both first access and re-access
- Evaluate algorithm by running it on a particular string of memory references (reference string) and computing the number of page faults on that string
  - String is just page numbers, not full addresses
  - Repeated access to the same page does not cause a page fault
- In all our examples, the reference string is

7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1

Graph of Page Faults Versus The Number of Frames



First-In-First-Out (FIFO) Algorithm

- Reference string:
  7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1

- 3 frames (3 pages can be in memory at a time per process)

1	7	2	4	0	7	
2	0	3	2	1	0	**15 page faults**
3	1	0	3	2	1	

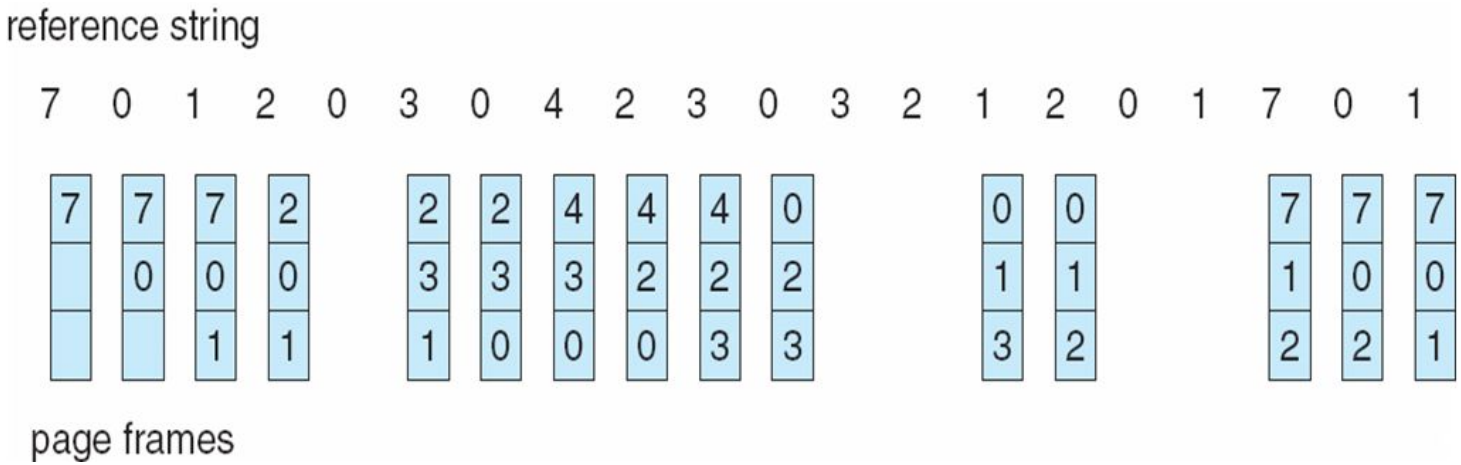
Belady's Anomaly

- Calculate the number of page fault for the following reference string using FIFO with the frame size as 3 & 4

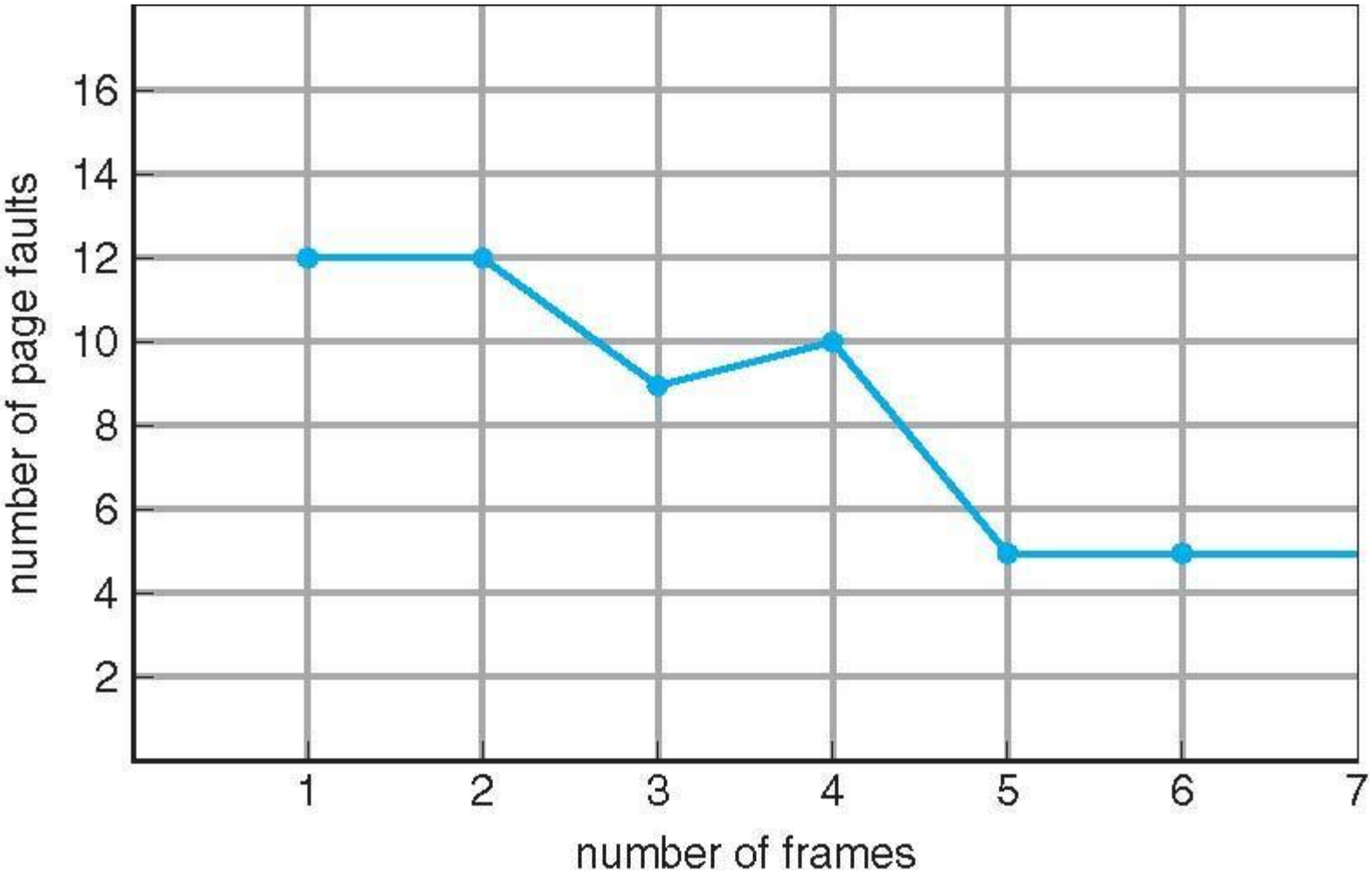
4 3 2 1 4 3 5 4 3 2 1 5

  - Adding more frames can cause more page faults!
  - This anomaly decreases the reliability of the replacement algorithm

FIFO Page Replacement



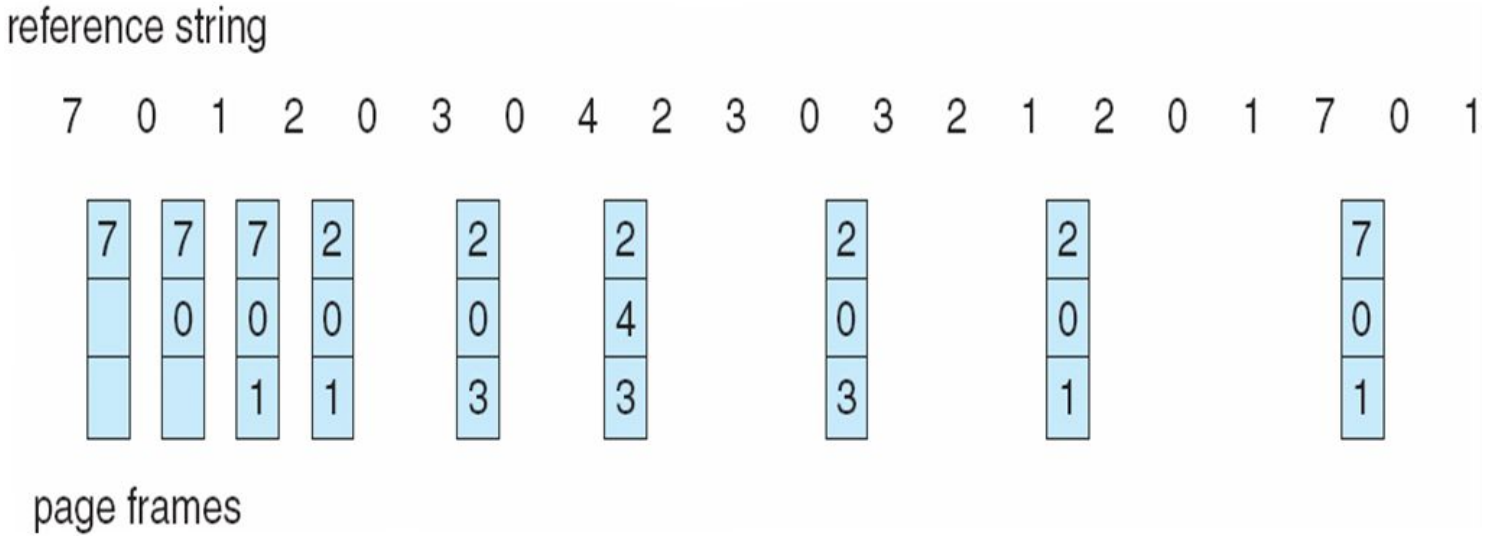
FIFO Illustrating Belady's Anomaly



Optimal Algorithm

- Replace page that will not be referenced for longest period of time.
- Produce minimal page fault.
- How do you know this?
  - Can't read the future
- Used for measuring how well your algorithm performs

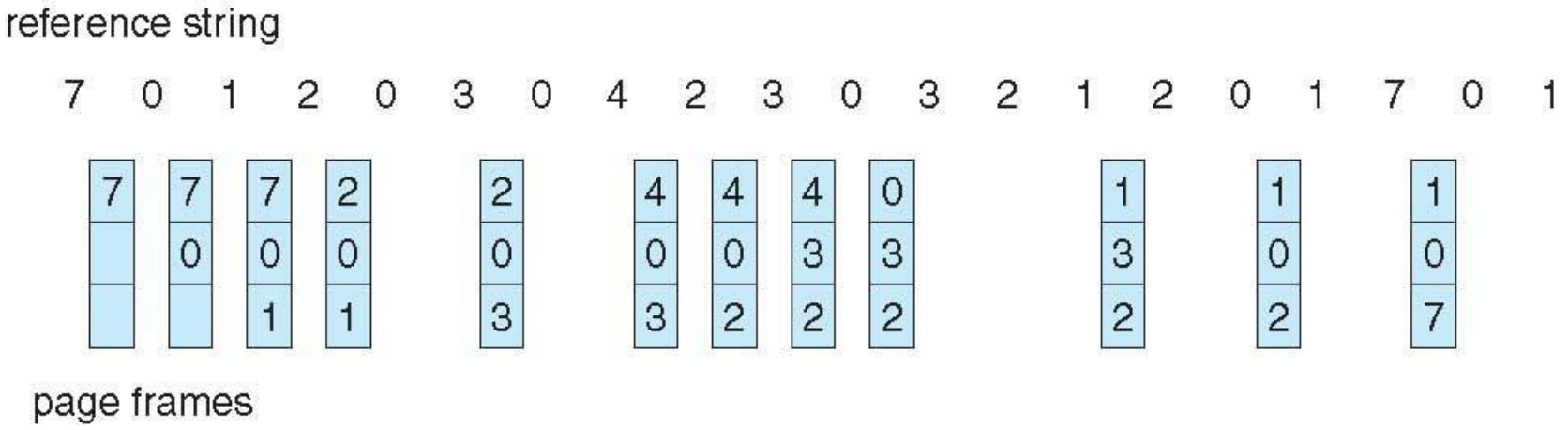
Optimal Page Replacement



- 9 page fault
- This algorithm provides the minimum number of page faults, it cannot be implemented
- There is no provision for the O.S to know the future page reference
- No practical use in page replacement

Least Recently Used (LRU) Algorithm

- Use past knowledge rather than future
- Replace page that has not been used in the most amount of time
- Associate time of last use with each page
- 12 faults – better than FIFO but worse than OPT
- Generally good algorithm and frequently used
- But how to implement?



THANK YOU