

# **CS241 System Programming Memory Management (V)**

Klara Nahrstedt

Lecture 32

4/12/2006



# Content

- Replacement Policies
  - LRU
  - NFU
  - Second Chance
  - Clock Page Algorithm
- Design Issues for Paging Systems
  - Frame allocation policies

# Administrative

- MP4 is posted, due April 17, 2006
- Quiz 9 is April 14, 2006
  - Topics for Quiz 9:
    - Basic memory management (T: Chapter 4.1)
    - Swapping (T: Chapter 4.2)
    - Paging (T: Chapter 4.3..1-4.3.3)

# LRU

12 references,  
10 faults

Page Refs	3 Page Frames			
	Fault?	Page Contents		
A	yes	A		
B	yes	B	A	
C	yes	C	B	A
D	yes	D	C	B
A	yes	A	D	C
B	yes	B	A	D
E	yes	E	B	A
A	no	A	E	B
B	no	B	A	E
C	yes	C	B	A
D	yes	D	C	B
E	yes	E	D	C

# Least Recently Used Issues

- Does not suffer from Belady's anomaly
- How to track “recency”?
  - use time
    - record time of reference with page table entry
    - use counter as clock
    - search for smallest time.
  - use stack
    - remove reference of page from stack (linked list)
    - push it on top of stack
- both approaches require large processing overhead, more space, and hardware support.

# LRU and Anomalies

Anomalies  
cannot  
occur,  
why?

12 references,  
8 faults

Page Refs	4 Page Frames				
	Fault?	Page Contents			
A	yes	A			
B	yes	B	A		
C	yes	C	B	A	
D	yes	D	C	B	A
A	no	A	D	C	B
B	no	B	A	D	C
E	yes	E	B	A	D
A	no	A	E	B	D
B	no	B	A	E	D
C	yes	C	B	A	E
D	yes	D	C	B	A
E	yes	E	D	C	B

# NFU: A LRU Approximation

- NFU (Not Frequently Used): Evict a page that is NOT frequently used;  
LRU: evict a page that is LEAST recently used;
- NFU Implementation: simpler than LRU
  - additional reference bits
  - a register is kept per page
  - a one bit is set in the register if the page is referenced
  - the register is shifted by one after some time interval
  - 00110011 would be accessed more recently than 00010111
  - the page with register holding the lowest number is the least recently used.
  - the value may not be unique. use FIFO to resolve conflicts.

# Second Chance

- Only one reference bit in the page table entry.
  - 0 initially
  - 1 When a page is referenced
- pages are kept in FIFO order using a linked list.
- Choose “victim” to evict
  - Select head of FIFO
  - If page has reference bit (R ) set, reset bit and select next page in FIFO list.
  - keep processing until reach page with zero reference bit and page that one out.

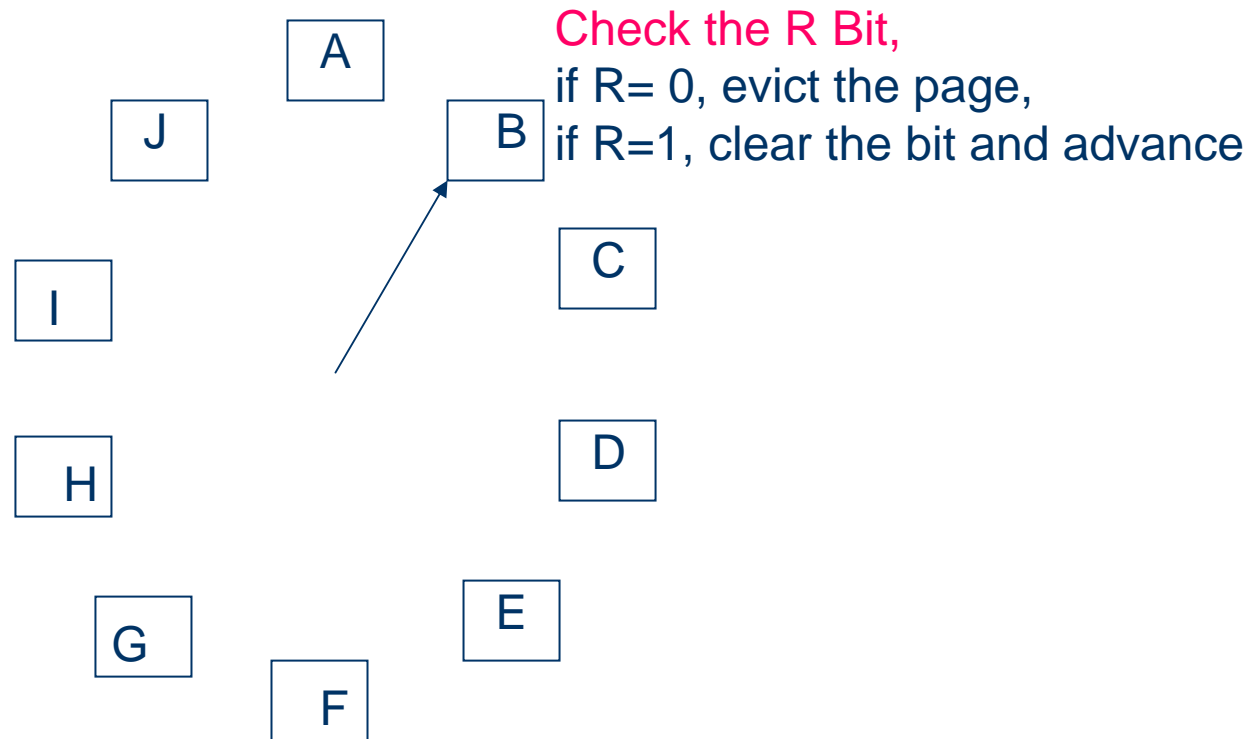
# Second Chance Example

12 references,  
9 faults

Page Refs	3 Page Frames			
	Fault?	Page Contents		
A	yes	A*		
B	yes	B*	A*	
C	yes	C*	B*	A*
D	yes	D*	C	B
A	yes	A*	D*	C
B	yes	B*	A*	D*
E	yes	E*	B	A
A	no	E*	B	A*
B	no	E*	B*	A*
C	yes	C*	E	B
D	yes	D*	C*	E
E	no	D*	C*	E*

# Clock Page Replacement Algorithm

- Similar to second chance, however
  - Keep all page frames on a **circular list** in the form of a clock



# Page Classes

- 1.(0,0) neither referenced nor dirtied
  - 2.(0,1) not referenced (recently) but dirtied
  - 3.(1,0) referenced but clean
  - 4.(1,1) referenced and dirtied
- select a page from lowest class
  - if conflict, use random or FIFO.

# Ad Hoc Techniques

- Separate Page Out from Page In.
  - Keep a pool of free frames.
  - When a page is to be replaced, use a free frame.
  - Read the faulting page and restart the faulting process while page out is occurring.
- Write dirty pages to disk whenever the paging device is free and reset the dirty bit. This allows page replacement algorithms to replace clean pages.
- Cache paged out pages in primary memory.
  - Page out dirty pages as before.
  - Return pages to a free pool but remember which page frame they are.
  - If system needs to map page in again, reuse page.
  - If system needs to page in data, choose any page in pool.
  - System V, R4 implements this strategy

# Frame Allocation: Minimum

- How are the page frames allocated to individual virtual memories in a multi-programmed environment?
- The simple case would be to allocate a minimum number of frames per process.
  - most instructions require two operands
  - include an extra page for paging out and one for paging in.
  - moves and indirection instructions might require more.

# Equal Allocation

- allocate an equal number of frames per job
  - but jobs use memory unequally
  - high priority jobs have same number of page frames and low priority jobs
  - degree of multiprogramming might vary

# Proportional Allocation

- allocate a number of frames per job proportional to job size
  - Challenge: how do you determine job size: by run command parameters or dynamically?

# Local versus Global Allocation Policies

- **Local allocation** policy uses only a fixed fraction of memory from process' own pages for allocating frames for each process.
  - Number of page frames assigned to each process is constant
- **Global allocation** policy uses page frames among all runnable processes
  - Number of page frames assigned to each process varies in time

# Summary

- Mostly used replacement algorithms in virtual memory systems can be found
  - second chance, clock page replacement, working set, and approximations of LRU
- Read Chapter T: 4.4.1, 4.4.2, 4.4.3, 4.4.4, 4.4.5, 4.4.6, and 4.5.1-4.5.3