CS 423
Operating System Design: Virtual Memory Wrap-Up

Professor Adam Bates
Spring 2018

Goals for Today

- Learning Objective:
  - Conclude exploration of virtual memory systems
- Announcements, etc:
  - Next C4 Summaries due today
  - Midterm March 7th!! Details on next slide
  - MP2 due March 16th
  - Strike Update: Classes continue as scheduled, assignment distribution/grading will likely be affected.

Reminder: Please put away devices at the start of class

Midterm Details

- In-Class on March 7th.
  - i.e., *50 minutes*
- Multiple choice
- 20-30 Questions
- **Openbook**: Textbooks, paper notes, printed sheets allowed. *No electronic devices permitted (or necessary)!*
- **Content**: All lecture and text material covered prior to March 5th (i.e., up to and including memory).
- We will have a review session, Q&A on March 5th.

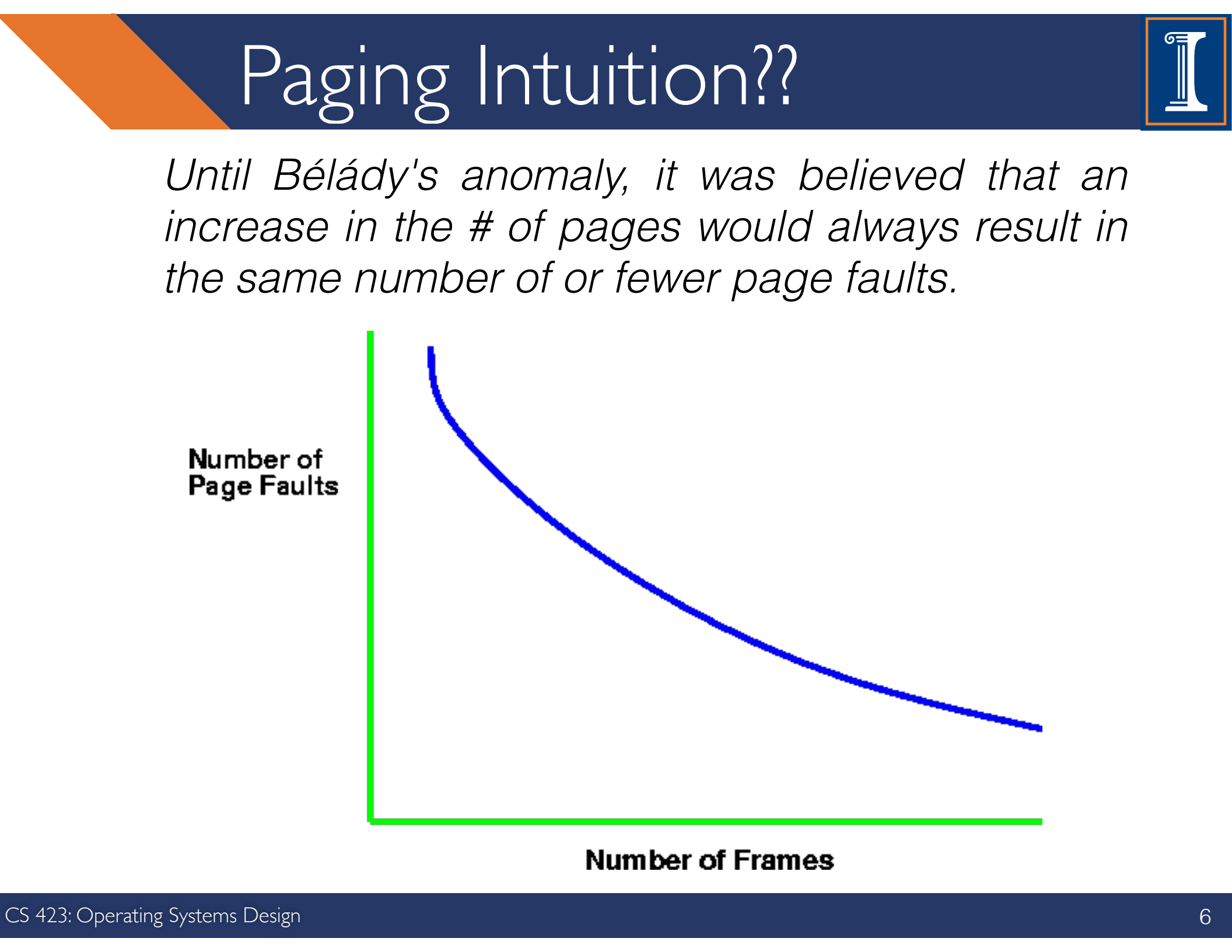
Page Replacement Strategies

- **The Principle of Optimality**
  - Replace the page that will not be used the most time in the future.
- **Random page replacement**
  - Choose a page randomly
- **FIFO - First in First Out**
  - Replace the page that has been in primary memory the longest
- **LRU - Least Recently Used**
  - Replace the page that has not been used for the longest time
- **LFU - Least Frequently Used**
  - Replace the page that is used least often
- **Second Chance**
  - An approximation to LRU.

Question from Last Class

Belady's Anomally...

Paging Intuition??



Until Bélády's anomaly, it was believed that an increase in the # of pages would always result in the same number of or fewer page faults.

FIFO w/ Page #'s 3 vs. 4

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

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	D	C	B	A
B	no	D	C	B	A
E	yes	E	D	C	B
A	yes	A	E	D	C
B	yes	B	A	E	D
C	yes	C	B	A	E
D	yes	D	C	B	A
E	yes	E	D	C	B

Belady's Anomaly

- **Increasing the number of page frames will affect the order in which items are removed.**
- For certain memory access patterns, this can actually increase the page fault rate!
- Belay's Anomaly is *reference string dependent*; intuition about increasing page count should hold in general case.

FIFO w/ Page #'s 3 vs. 4

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

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	D	C	B	A
B	no	D	C	B	A
E	yes	E	D	C	B
A	yes	A	E	D	C
B	yes	B	A	E	D
C	yes	C	B	A	E
D	yes	D	C	B	A
E	yes	E	D	C	B

FIFO w/ Page #'s 3 vs. 4

Page Refs	3 Page Frames			
	Fault?	Page Contents		
A	yes	A		
B	yes	B	A	
C	yes	C	B	
D	yes	D	C	
A	yes	A	D	
B	yes	B	A	
E	yes	E	B	
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

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

$\text{Cache}_3 \nsubseteq \text{Cache}_4$!!!!

Question from Last Class

Why not use very large pages to reduce page faults?

Paging Terminology

- **Reference string**: the memory reference sequence generated by a program.
- **Paging** – moving pages to (from) disk
- **Optimal** – the best (theoretical) strategy
- **Eviction** – throwing something out
- **Pollution** – bringing in useless pages/lines

Page Size Considerations

- Small pages
  - Reason:
    - Locality of reference tends to be small (256)
    - Less fragmentation
  - Problem: require large page tables
- Large pages
  - Reason
    - Small page table
    - I/O transfers have high seek time, so better to transfer more data per seek
  - Problem: Internal fragmentation, needless caching

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 circular list.
- Choose "victim" to evict
  - Select head of FIFO
  - If page has reference bit set, reset bit and select next page in FIFO list.
  - keep processing until you reach page with zero reference bit and page that one out.
- System V uses a variant of second chance

Second change is a variant of FIFO, not LRU.

LRU

12 references,
10 faults

Pro: (In spite of example,) provides near-optimal performance.

Con: costs of maintaining access history.

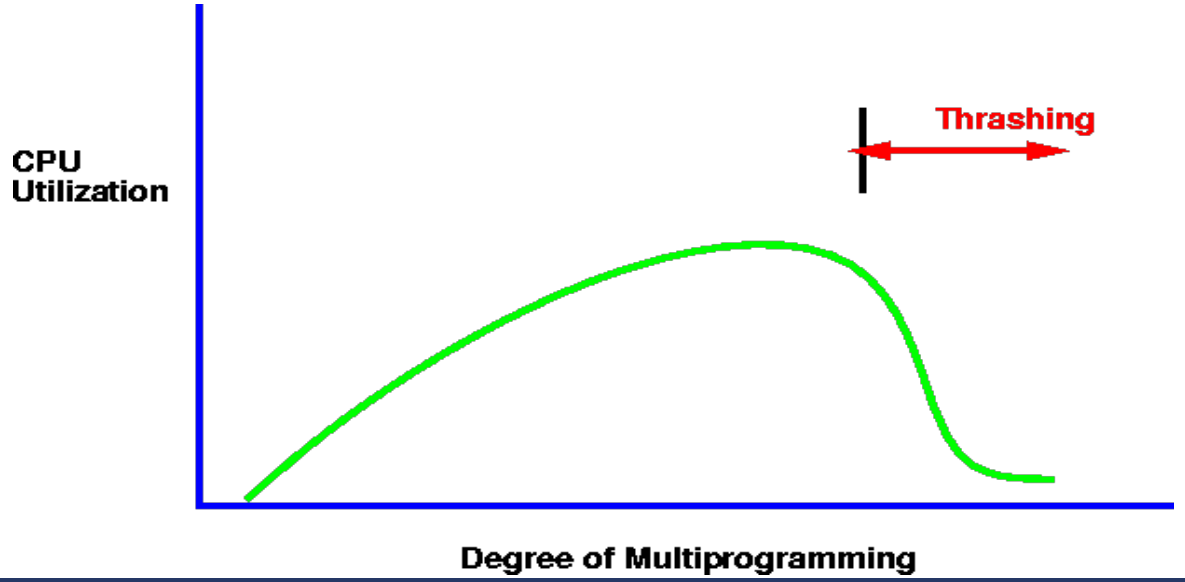
Page Refs	3 Page Frames			
	Fault?	Page Contents		
A	yes	A		
B	yes	B	A	
C	yes	C	B	A
D	yes	D	C	B
A	yes	A	D	C
B	yes	B	A	D
E	yes	E	B	A
A	no	A	E	B
B	no	B	A	E
C	yes	C	B	A
D	yes	D	C	B
E	yes	E	D	C

Thrashing

- Computations have locality.
  - **principle of locality**: a program clusters its access to data and text temporally.
- As page frames decrease, the page frames available are not large enough to contain the locality of the process.
- The processes start faulting heavily.
- Pages that are repeatedly read in, used and then immediately paged out.

Thrashing & CPU Utilization

- As the page rate goes up, processes get suspended on page out queues for the disk.
- the system may try to optimize performance by starting new jobs.
- starting new jobs will reduce the number of page frames available to each process, increasing the page fault requests.
- system throughput plunges.



Working Set

- the working set model accounts for locality.
- **Working Set**: the collection of pages that a process is working with, and which must thus be resident if the process is to avoid thrashing
  - Identify working set based on the pages referenced by process in last **t** seconds.
  - Do not schedule process unless entire working set is in main memory
- As the number of page frames increases above some threshold, the page fault rate will drop dramatically.