

Lecture 16: Memory Management

Page replacement algorithms

Recall: Paging

- Allocate VA & PA memory in **chunks of the same, fixed size** (**pages** and **frames**, respectively)
- Adjacent pages in VA need not map to contiguous frames in PA!
 - no external fragmentation!
 - possible **internal** fragmentation
 - when memory needs are not a multiple of a page
 - typical size of page/frame: 4KB to 16KB

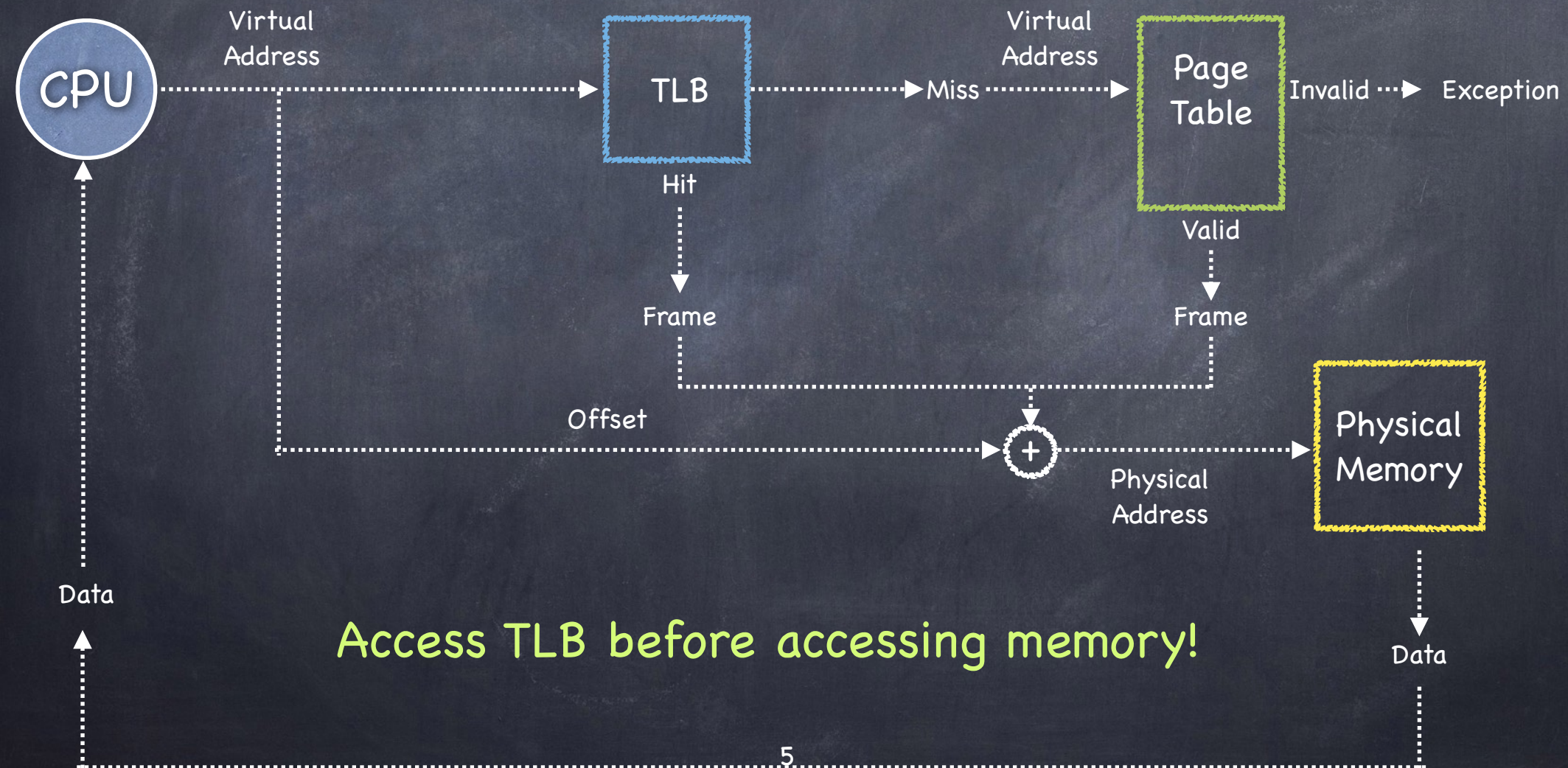
Recall: Basic goals in paging

- ◉ **Minimize Storage overhead**
 - **data structure overhead** (the Page Table itself)
 - **fragmentation**
 - ▶ How large should a page be?
- ◉ **Fast Address translation**
 - We need “fast” lookups on page table
- ◉ **Efficient sharing of physical memory**
 - By multiple processes

Recall: Where are we?

- Storage overheads
 - Minimized! Using multi-level page tables.
- How about address translation time?
 - Every new level of paging
 - ▶ lower storage overhead, but higher translation time
 - ▶ Solution: cache address translation results in Translation Lookaside Buffer (TLB)
 - Cache virtual address → frame mapping!

Recall: Address Translation with TLB



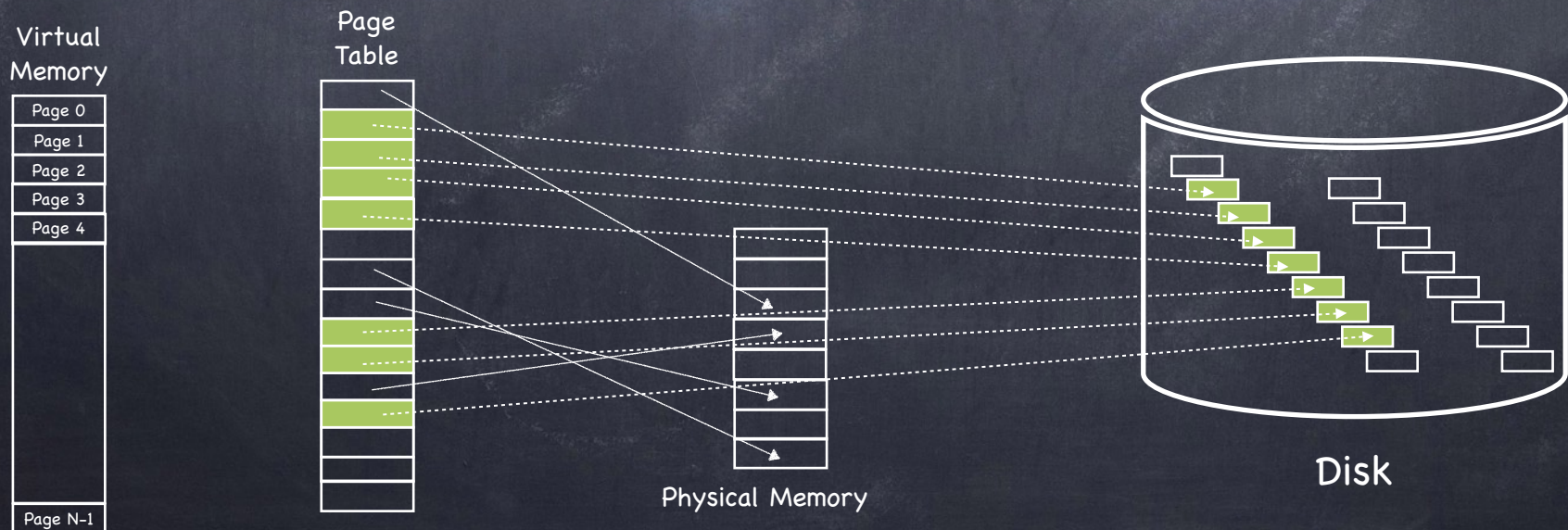
Recall:

Handling TLB Misses

- Hardware-managed (e.g., x86)
 - The hardware does the **page walk**
 - Hardware fetches PTE and inserts it in TLB
 - ▶ If TLB is full, must replace another TLB entry
 - Done transparently to system software
- Software-managed (e.g., MIPS)
 - Hardware raises an exception
 - OS does the **page walk**, fetches PTE, and inserts evicts entries in TLB

Recall: Virtual Memory

- Each process has the illusion of a large address space
 - 2^x bytes for x -bit addressing
- However, physical memory is usually much smaller
 - and we want to run multiple processes concurrently
- How do we give this illusion to multiple processes?
 - Virtual Memory: back every memory address with a file on disk



Recall: A Virtual Page can be...

- **Mapped** (present bit set in PTE) may trigger Page Fault
 - to a physical frame, with certain r/w/x permissions

- **Not mapped** (present bit not set in PTE)

Page
Fault

- in some physical frame, but not currently mapped
- or still in the original program file
- or needing to be zero-filled
- or on backing store (paged or swapped out)
- or not part of one of the processes' segment
 - ▶ Segmentation Fault!

When a page must be brought in...

- ① Find a free frame
 - Page replacement: evict a page if there are no free frames
- ① Issue disk request to fetch data for page
- ① Move "current process" to disk queue
- ① Context switch to new process
- ① Update PTE when disk completes
 - frame number, present bit, RWX bits, etc.
- ① Move "current process" to ready queue

Questions?

Page Replacement

- When physical memory is full, we need to choose a “victim” to evict
- Local vs Global replacement
 - **Local**: victim chosen from frames of process experiencing page fault
 - ▶ fixed allocation of frames per process
 - **Global**: victim chosen from frames allocated to any process
 - ▶ variable allocation of frames per process
- Goal: minimizing number of page faults

Page Replacement Algorithms

- **Random:** Pick any page to eject at random
 - Used mainly for comparison
- **FIFO:** The page brought in earliest is evicted
 - Ignores usage
- **LRU:** Evict page not been used the longest
 - Assumes past is good predictor of the future
- **MRU:** Evict most recently used page
 - Good for data accessed only once, e.g., a movie
- **LFU:** Evict least frequently used page
- **OPT:** Belady's algorithm

How do we pick a victim?

- We want:

- low page fault-rate
- page faults as inexpensive as possible

- We need:

- a way to compare the relative performance of different page replacement algorithms
- some absolute notion of what a “good” page replacement algorithm should accomplish

Comparing Page Replacement Algorithms


- Record a trace of the pages accessed by a process
 - E.g. 3,1,4,2,5,2,1,2,3,4 (or c,a,d,b,e,b,a,b,c,b)
- Simulate behavior of page replacement algorithm on trace
- Record number of page faults generated

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | | | | | | | | | | | | |
| Page Frames | 0 | | | | | | | | | | | | |
| | 1 | | | | | | | | | | | | |
| | 2 | | | | | | | | | | | | |
| Faults | X | | | | | | | | | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | | | | | | | | | | | |
| Page Frames | 0 | a | | | | | | | | | | | |
| | 1 | | | | | | | | | | | | |
| | 2 | | | | | | | | | | | | |
| Faults | X | X | | | | | | | | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | | | | | | | | | | |
| Page Frames | 0 | a | a | | | | | | | | | | |
| | 1 | | b | | | | | | | | | | |
| | 2 | | | | | | | | | | | | |
| Faults | X | X | X | | | | | | | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | a | a | a | | | | | | | | | |
| | 1 | | b | b | | | | | | | | | |
| | 2 | | | c | | | | | | | | | |
| Faults | X | X | X | X | | | | | | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | | | | | | | | |
| Page Frames | 0 | a | a | a | a | | | | | | | | |
| | 1 | | b | b | b | | | | | | | | |
| | 2 | | | | c | d | | | | | | | |
| Faults | X | X | X | X | ✓ | | | | | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | | | | | | | |
| Page Frames | 0 | a | a | a | a | a | | | | | | | |
| | 1 | | b | b | b | b | | | | | | | |
| | 2 | | | | c | d | d | | | | | | |
| Faults | X | X | X | X | ✓ | ✓ | | | | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | a | a | a | a | a | a | | | | | | |
| | 1 | | b | b | b | b | b | | | | | | |
| | 2 | | | c | d | d | d | | | | | | |
| Faults | X | X | X | X | ✓ | ✓ | X | | | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | | | | | |
| Page Frames | 0 | a | a | a | a | a | a | a | | | | | |
| | 1 | | b | b | b | b | b | b | | | | | |
| | 2 | | | c | d | d | d | e | | | | | |
| Faults | X | X | X | X | ✓ | ✓ | X | ✓ | | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | | | | |
| Page Frames | 0 | a | a | a | a | a | a | a | a | | | | |
| | 1 | | b | b | b | b | b | b | b | | | | |
| | 2 | | | c | d | d | d | e | e | | | | |
| Faults | X | X | X | X | ✓ | ✓ | X | ✓ | ✓ | | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | | a | a | a | a | a | a | a | a | | | |
| | 1 | | | b | b | b | b | b | b | b | | | |
| | 2 | | | | c | d | d | d | e | e | e | | |
| Faults | X | X | X | X | ✓ | ✓ | X | ✓ | ✓ | X | | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | a | a | a | a | a | a | a | a | a | c | | |
| | 1 | | b | b | b | b | b | b | b | b | b | | |
| | 2 | | | c | d | d | d | e | e | e | e | | |
| Faults | X | X | X | X | ✓ | ✓ | X | ✓ | ✓ | X | X | | |

Process can use 3 frames
(3 pages in memory)


 Page loaded

Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | a | a | a | a | a | a | a | a | a | c | c | |
| | 1 | | b | b | b | b | b | b | b | b | b | d | |
| | 2 | | | c | d | d | d | e | e | e | e | e | |
| Faults | X | X | X | X | ✓ | ✓ | X | ✓ | ✓ | X | X | ✓ | |

Process can use 3 frames
(3 pages in memory)

 Page loaded


Optimal Page Replacement

- Replace page needed furthest in future

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | a | a | a | a | a | a | a | a | a | c | c | c |
| | 1 | | b | b | b | b | b | b | b | b | b | d | d |
| | 2 | | | | c | d | d | d | e | e | e | e | e |
| Faults | X | X | X | X | ✓ | ✓ | X | ✓ | ✓ | X | X | ✓ | |

7 page faults

Process can use 3 frames
(3 pages in memory)

 Page loaded

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | a | a | a | a | a | a | a | a | a | a | d | d |
| | 1 | | b | b | b | b | b | b | b | b | b | a | e |
| | 2 | | | c | c | c | c | c | c | c | c | b | b |
| | 3 | | | | d | d | d | e | e | e | e | e | c |
| Faults | X | X | X | X | ✓ | ✓ | X | ✓ | ✓ | ✓ | X | ✓ | |

6 page faults


Process can use 4 frames
(4 pages in memory)

FIFO Replacement

- Replace pages in the order they come into memory

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | | a | a | a | d | d | d | e | e | e | e | e |
| | 1 | | | b | b | b | a | a | a | a | c | c | c |
| | 2 | | | | c | c | c | b | b | b | b | d | d |
| Faults | X | X | X | X | X | X | X | ✓ | ✓ | X | X | ✓ | |

Process can use 3 frames
(3 pages in memory)

 Page loaded

9 page faults

FIFO Replacement

- Replace pages in the order they come into memory

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | a | a | a | a | a | a | e | e | e | e | d | d |
| | 1 | | b | b | b | b | b | b | a | a | a | a | e |
| | 2 | | | c | c | c | c | c | c | b | b | b | b |
| | 3 | | | | d | d | d | d | d | d | c | c | c |
| Faults | X | X | X | X | ✓ | ✓ | X | X | X | X | X | X | |

Process can use 4 frames
(4 pages in memory)

 Page loaded

10 page faults

More frames → more page faults?

Belady's Anomaly

Locality of Reference

- If a process access a memory location, then it is likely that
 - the same memory location is going to be accessed again in the near future (temporal locality)
 - nearby memory locations are going to be accessed in the future (spatial locality)
- 90% of the execution of a program is sequential
- Most iterative constructs consist of a relatively small number of instructions

LRU: Least Recently Used

- Replace page not referenced for the longest time

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | | | | | | | |
| Page Frames | 0 | a | a | a | a | a | a | | | | | | |
| | 1 | | b | b | b | b | b | | | | | | |
| | 2 | | | c | c | c | c | | | | | | |
| | 3 | | | | d | d | d | | | | | | |
| Faults | X | X | X | X | ✓ | ✓ | | | | | | | |

Process can use 4 frames
(4 pages in memory)

LRU: Least Recently Used

- Replace page not referenced for the longest time

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | | | | | | |
| Page Frames | 0 | a | a | a | a | a | a | | | | | | |
| | 1 | | b | b | b | b | b | | | | | | |
| | 2 | | | c | c | c | c | | | | | | |
| | 3 | | | | d | d | d | | | | | | |
| Faults | X | X | X | X | ✓ | ✓ | X | | | | | | |

Process can use 4 frames
(4 pages in memory)

LRU: Least Recently Used

- Replace page not referenced for the longest time

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | | | | | | |
| Page Frames | 0 | a | a | a | a | a | a | a | | | | | |
| | 1 | | b | b | b | b | b | b | | | | | |
| | 2 | | | c | c | c | c | e | | | | | |
| | 3 | | | | d | d | d | d | | | | | |
| Faults | X | X | X | X | ✓ | ✓ | X | | | | | | |

Process can use 4 frames
(4 pages in memory)

LRU: Least Recently Used

- Replace page not referenced for the longest time

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | e | a | b | c | d | e | |
| Page Frames | 0 | | a | a | a | a | a | a | a | a | a | a | e |
| | 1 | | | b | b | b | b | b | b | b | b | b | b |
| | 2 | | | | c | c | c | c | e | e | e | e | d |
| | 3 | | | | | d | d | d | d | d | d | c | c |
| Faults | X | X | X | X | ✓ | ✓ | X | ✓ | ✓ | X | X | X | |

8 page faults

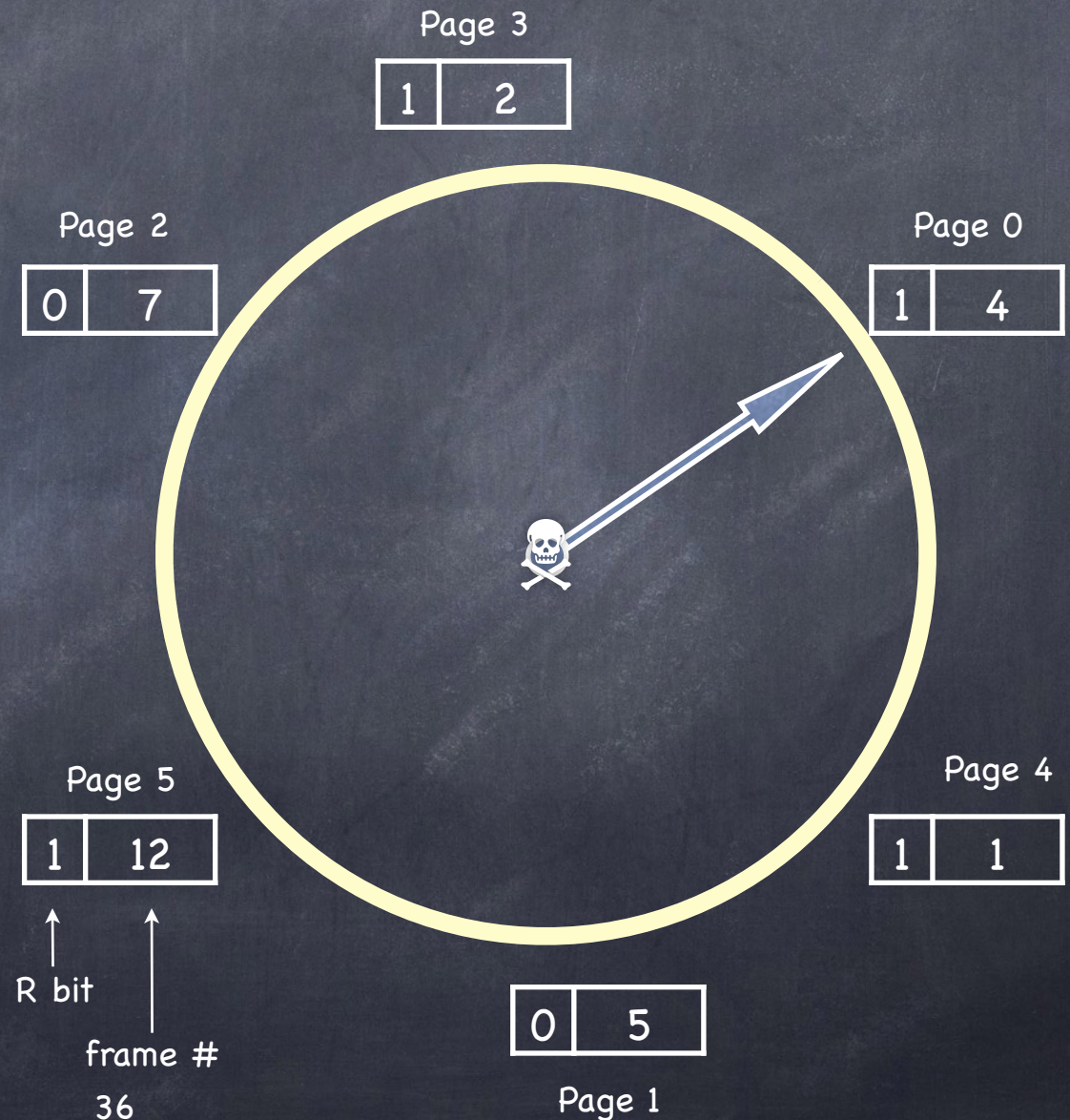
Process can use 4 frames
(4 pages in memory)

Implementing LRU

- On **reference**: timestamp each page
- On **eviction**: scan for oldest page
- Problems:
 - Large page lists
 - Timestamps are costly
- Solution: **approximate LRU**
 - after all, LRU is already an approximation! (of OPT)
 - Next lecture

The Clock Algorithm

- Organize pages in memory as a circular list
- When page is referenced, set its reference bit R to 1
- On page fault
 - if $R = 1$: set $R=0$
 - else if $R = 0$:
 - evict page pointed to
 - load new page
 - set R to 1
 - advance hand



Clock Page Replacement

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|
| Trace | c | a | d | b | | | | | | | |
| Page Frames | 0 | a | a | a | a | | | | | | |
| | 1 | b | b | b | b | | | | | | |
| | 2 | c | c | c | c | | | | | | |
| | 3 | d | d | d | d | | | | | | |
| Faults | | | | | | | | | | | |

Page table entries
for resident pages

Hand clock:

| | |
|---|---|
| 1 | a |
| 1 | b |
| 1 | c |
| 1 | d |

Clock Page Replacement

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|
| Trace | c | a | d | b | e | | | | | | |
| Page Frames | 0 | a | a | a | a | a | | | | | |
| | 1 | b | b | b | b | b | | | | | |
| | 2 | c | c | c | c | c | | | | | |
| | 3 | d | d | d | d | d | | | | | |
| Faults | | | | | X | | | | | | |

Page table entries
for resident pages

Hand clock:

| | |
|---|---|
| 1 | a |
| 1 | b |
| 1 | c |
| 1 | d |

Clock Page Replacement

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|
| Trace | c | a | d | b | e | b | | | | | |
| Page Frames | 0 | a | a | a | a | e | | | | | |
| | 1 | b | b | b | b | b | | | | | |
| | 2 | c | c | c | c | c | | | | | |
| | 3 | d | d | d | d | d | | | | | |
| Faults | | | | | X | | | | | | |

Page table entries
for resident pages

Hand clock:

| | |
|---|---|
| 1 | a |
| 1 | b |
| 1 | c |
| 1 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 0 | c |
| 0 | d |

Clock Page Replacement

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|
| Trace | c | a | d | b | e | b | a | | | | |
| Page Frames | 0 | a | a | a | a | e | e | | | | |
| | 1 | b | b | b | b | b | b | | | | |
| | 2 | c | c | c | c | c | c | | | | |
| | 3 | d | d | d | d | d | d | | | | |
| Faults | | | | | X | | X | | | | |

Page table entries
for resident pages

Hand clock:

| | |
|---|---|
| 1 | a |
| 1 | b |
| 1 | c |
| 1 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 0 | c |
| 0 | d |

Clock Page Replacement

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|
| Trace | c | a | d | b | e | b | a | b | | | |
| Page Frames | 0 | a | a | a | a | e | e | e | | | |
| | 1 | b | b | b | b | b | b | b | | | |
| | 2 | c | c | c | c | c | c | a | | | |
| | 3 | d | d | d | d | d | d | d | | | |
| Faults | | | | | X | | X | | | | |

Page table entries
for resident pages

Hand clock:

| | |
|---|---|
| 1 | a |
| 1 | b |
| 1 | c |
| 1 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 1 | a |
| 0 | d |

Clock Page Replacement

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|
| Trace | c | a | d | b | e | b | a | b | c | | |
| Page Frames | 0 | a | a | a | a | e | e | e | e | | |
| | 1 | b | b | b | b | b | b | b | b | | |
| | 2 | c | c | c | c | c | c | a | a | | |
| | 3 | d | d | d | d | d | d | d | d | | |
| Faults | | | | | X | | X | | X | | |

Page table entries for resident pages

Hand clock:

| | |
|---|---|
| 1 | a |
| 1 | b |
| 1 | c |
| 1 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 1 | a |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 1 | a |
| 0 | d |

Clock Page Replacement

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|
| Trace | c | a | d | b | e | b | a | b | c | d | |
| Page Frames | 0 | a | a | a | a | e | e | e | e | e | |
| | 1 | b | b | b | b | b | b | b | b | b | |
| | 2 | c | c | c | c | c | c | a | a | a | |
| | 3 | d | d | d | d | d | d | d | d | c | |
| Faults | | | | | X | | X | | X | X | |

Page table entries
for resident pages

Hand clock:

| | |
|---|---|
| 1 | a |
| 1 | b |
| 1 | c |
| 1 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 1 | a |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 1 | a |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 1 | a |
| 1 | c |

Clock Page Replacement

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------|---|---|---|---|---|---|---|---|---|---|----|
| Trace | c | a | d | b | e | b | a | b | c | d | |
| Page Frames | 0 | a | a | a | a | e | e | e | e | e | d |
| | 1 | b | b | b | b | b | b | b | b | b | b |
| | 2 | c | c | c | c | c | c | a | a | a | a |
| | 3 | d | d | d | d | d | d | d | d | c | c |
| Faults | | | | | X | | X | | X | X | |

Page table entries
for resident pages

Hand clock:

| | |
|---|---|
| 1 | a |
| 1 | b |
| 1 | c |
| 1 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 0 | c |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 0 | b |
| 1 | a |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 1 | a |
| 0 | d |

| | |
|---|---|
| 1 | e |
| 1 | b |
| 1 | a |
| 1 | c |

| | |
|---|---|
| 1 | d |
| 0 | b |
| 0 | a |
| 0 | c |

Local vs. Global Page Replacement

- Local: processes select victim among allocated frames allocated to them
 - Can lead to under utilization
- Global: Select any free frame, even if allocated to another process
 - Processes lose control over their own page fault rate

Brother, can you spare a frame?

FIFO

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|--------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | c | d | a | b | c | d | |
| 0 | | a | a | a | d | d | d | c | c | c | b | b | b |
| 1 | | | b | b | b | a | a | a | d | d | d | c | c |
| 2 | | | | c | c | c | b | b | b | a | a | a | d |
| Faults | X | X | X | X | X | X | X | X | X | X | X | X | X |

Brother, can you spare a frame?

FIFO

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|--------|---|---|---|---|---|---|---|---|---|---|----|----|----|
| Trace | a | b | c | d | a | b | c | d | a | b | c | d | |
| 0 | | a | a | a | a | a | a | a | a | a | a | a | a |
| 1 | | | b | b | b | b | b | b | b | b | b | b | b |
| 2 | | | | c | c | c | c | c | c | c | c | c | c |
| 3 | | | | | d | d | d | d | d | d | d | d | d |
| Faults | X | X | X | X | X | | | | | | | | |

So, what's wrong with global replacement?

Sharing Memory as a Cache

- Demand paging enables frames to cache part of a process VA space
- If the cache is large enough, hit ratio is high
 - few page faults
- What if there aren't enough frames to go around?
 - should **decrease** degree of multiprogramming
 - ▶ swapped out process can then release its frames

Instead...

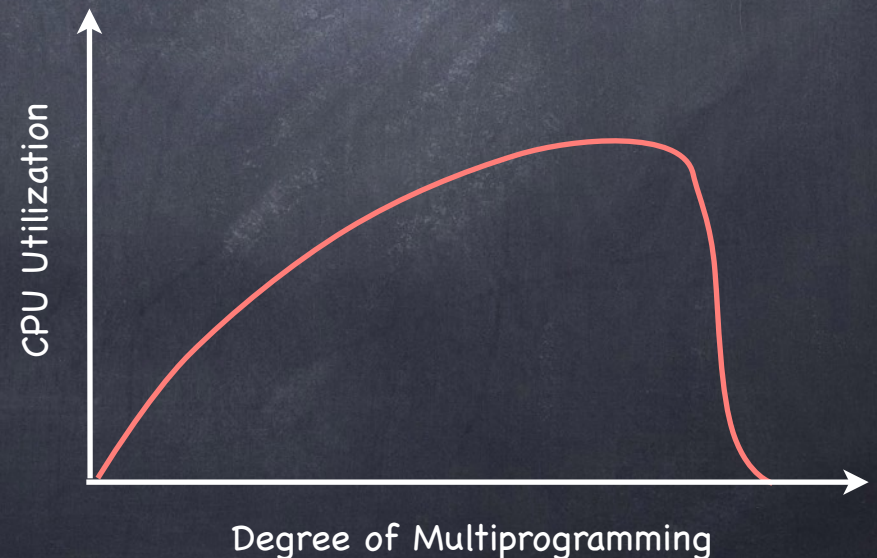
- ◉ When not enough frames...
 - high page fault rate
 - low CPU utilization
 - OS may **increase** degree of multiprogramming!

Instead...

- When not enough frames...
 - high page fault rate
 - low CPU utilization
 - OS may **increase** degree of multiprogramming!

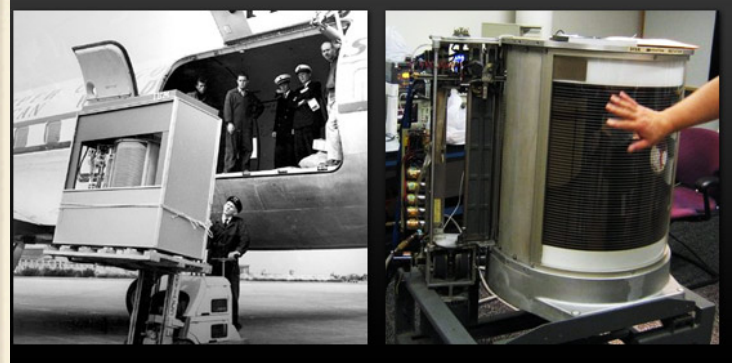
- **Thrashing**

- process spends all its time swapping pages in and out



Why “thrashing”?

“THRASH” DATES FROM THE 1960’S, WHEN DISK DRIVES WERE AS LARGE AS WASHING MACHINES. IF A PROGRAM’S WORKING SET DID NOT FIT IN MEMORY, THE SYSTEM WOULD NEED TO SHUFFLE MEMORY PAGES BACK AND FORTH TO DISK. THIS BURST OF ACTIVITY WOULD VIOLENTLY SHAKE THE DISK DRIVE.



The first hard disk drive—the IBM Model 350 Disk File (came w/IBM 305 RAMAC, 1956).

Total storage =
5 million characters (just
under 5 MB).

Locality of Reference

- If a process access a memory location, then it is likely that
 - the same memory location is going to be accessed again in the near future (temporal locality)
 - nearby memory locations are going to be accessed in the future (spatial locality)

Tracking Locality

- When a process executes it moves from **locality** (set of pages used together) to **locality**
 - the size of the process' locality (a.k.a. its **working set**) can change over time
- **Goal:** track the size of the process' working set, dynamically acquiring and releasing frames as necessary

The Working Set Model

- Choose Δ page references as **WS sliding window**
 - track WS for the last Δ page references
- $WSS_i = \#$ of distinct pages referenced by p_i in latest Δ references
 - Δ too small does not cover locality
 - Δ too large covers many localities
- Thrashing if $\sum_i WSS_i > \#$ frames
 - if so, swap out one of the processes; free its frames
- If enough free frames, increase degree of multiprogramming

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | ■ | ■ | | | | | | | | | |
| | Page b | | | | | | | | | | | | | |
| | Page c | | | | ■ | ■ | | | | | | | | |
| | Page d | | | ■ | ■ | ■ | | | | | | | | |
| | Page e | | ■ | ■ | ■ | ■ | | | | | | | | |
| Faults | × | × | × | × | | | | | | | | | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | | | | | | | | |
| | Page b | | | | | | | | | | | | | |
| | Page c | | | | █ | █ | | | | | | | | |
| | Page d | | | █ | █ | █ | | | | | | | | |
| | Page e | | █ | █ | █ | █ | | | | | | | | |
| Faults | × | × | × | × | ✓ | | | | | | | | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | █ | | | | | | | |
| | Page b | | | | | | | | | | | | | |
| | Page c | | | | █ | █ | █ | | | | | | | |
| | Page d | | | █ | █ | █ | █ | | | | | | | |
| | Page e | | █ | █ | █ | █ | | | | | | | | |
| Faults | × | × | × | × | ✓ | ✓ | | | | | | | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | █ | | | | | | | |
| | Page b | | | | | | | █ | | | | | | |
| | Page c | | | | █ | █ | █ | █ | | | | | | |
| | Page d | | | █ | █ | █ | █ | █ | | | | | | |
| | Page e | | █ | █ | █ | █ | | | | | | | | |
| Faults | × | × | × | × | ✓ | ✓ | × | | | | | | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | █ | | | | | | | |
| | Page b | | | | | | | █ | █ | | | | | |
| | Page c | | | | █ | █ | █ | █ | █ | | | | | |
| | Page d | | | █ | █ | █ | █ | █ | █ | | | | | |
| | Page e | | █ | █ | █ | █ | | | | | | | | |
| Faults | × | × | × | × | ✓ | ✓ | × | ✓ | | | | | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | █ | | | | | | | |
| | Page b | | | | | | | █ | █ | █ | | | | |
| | Page c | | | | █ | █ | █ | █ | █ | █ | | | | |
| | Page d | | | █ | █ | █ | █ | █ | █ | █ | | | | |
| | Page e | | █ | █ | █ | | | | | | █ | | | |
| Faults | × | × | × | × | ✓ | ✓ | × | ✓ | × | | | | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | █ | | | | | | | |
| | Page b | | | | | | | █ | █ | █ | █ | | | |
| | Page c | | | | █ | █ | █ | █ | █ | █ | █ | | | |
| | Page d | | | █ | █ | █ | █ | █ | █ | █ | █ | | | |
| | Page e | | █ | █ | █ | █ | | | | | █ | █ | | |
| Faults | × | × | × | × | ✓ | ✓ | × | ✓ | × | ✓ | | | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | █ | | | | | | | |
| | Page b | | | | | | | █ | █ | █ | █ | | | |
| | Page c | | | | █ | █ | █ | █ | █ | █ | █ | █ | | |
| | Page d | | | █ | █ | █ | █ | █ | █ | █ | █ | | | |
| | Page e | | █ | █ | █ | █ | | | | | █ | █ | | |
| Faults | × | × | × | × | ✓ | ✓ | × | ✓ | × | ✓ | ✓ | | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | █ | | | | | | █ | |
| | Page b | | | | | | | █ | █ | █ | | | | |
| | Page c | | | | █ | █ | █ | █ | █ | █ | █ | █ | █ | |
| | Page d | | | █ | █ | █ | █ | █ | █ | █ | | | | |
| | Page e | | █ | █ | █ | | | | | | █ | █ | █ | |
| Faults | × | × | × | × | ✓ | ✓ | × | ✓ | × | ✓ | ✓ | × | | |

WS Page Replacement

$$\Delta = 4$$

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|
| Trace | e | d | a | c | c | d | b | c | e | c | e | a | d | |
| Pages in Memory | Page a | | | █ | █ | █ | █ | | | | | | █ | █ |
| | Page b | | | | | | | █ | █ | █ | █ | | | |
| | Page c | | | | █ | █ | █ | █ | █ | █ | █ | █ | █ | █ |
| | Page d | | | █ | █ | █ | █ | █ | █ | █ | | | | █ |
| | Page e | | █ | █ | █ | | | | | | █ | █ | █ | █ |
| Faults | × | × | × | × | ✓ | ✓ | × | ✓ | × | ✓ | ✓ | × | × | |

Where are we in CS4410?

- Threads and Processes
- CPU schedulers
- Concurrency / synchronization
- Address translation
- Storage I/O
- Network I/O

