CPSC313: Computer Hardware and Operating Systems

Unit 5: Virtual Memory

VM Wrap Up, 2-Handed Clock, and Q&A

Administrivia

- Last day of the term!
  - Don’t miss your Quiz 5 time!
  - Don’t miss your final exam time!
  - Exam period office hours are pinned on Piazza; take advantage to ask your questions!

Today

- Roadmap:
  - Virtual Memory Review/Summary
  - Extend the clock algorithm (from pre-class) to the 2-handed version
- Learning Objectives
  - Extend the clock algorithm to avoid having to write dirty pages during replacement.
  - Think about connections across the course!

The Big Picture

- What is the purpose of virtual memory?

The Big Picture

- What is the purpose of virtual memory?

  Process isolation: One process's address space has no effect on another's, unless those two processes explicitly set up communication.

- How does VM work?

The Big Picture

- What is the purpose of virtual memory?

  Process isolation: One process's address space has no effect on another's, unless those two processes explicitly set up communication.

- How does VM work?
  - Hardware/software partnership.
  - Hardware translates whatever addresses it can (TLB + for x86-64, page table). (Some processors have only a TLB.)
  - Software takes over where the hardware leaves off: page faults!

Other Benefits of Virtual Memory

- Processes' address spaces can be **larger than physical memory**.
  - Corollary: we can run a process even if some of its pages are not in memory.
- We can run a collection of processes even if the sum of the sizes of their (in-use) address spaces do not fit in memory.
- We call the loading of VM pages as they are needed demand paging.
- Main memory (DRAM or RAM) acts like a cache for the sum total of all processes' address spaces.
  - But if memory is a cache, we need replacement policies…

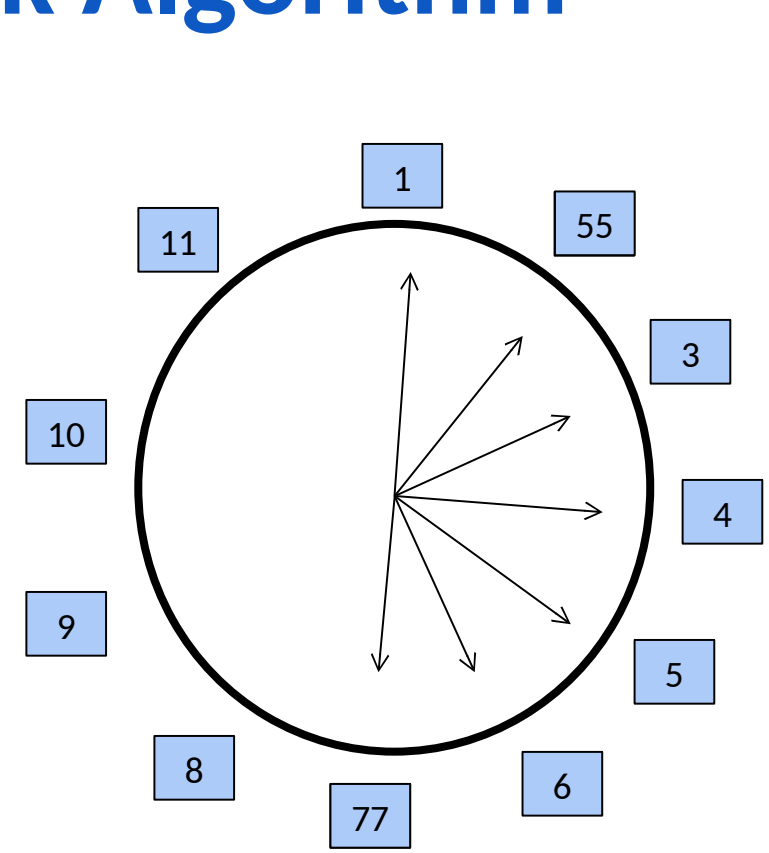
Why Clock and Not LRU?

Why Clock and Not LRU?

- The OS (which handles page replacement) does not have complete visibility into the access stream, so it cannot identify the LRU page.
  - Virtual memory pages can be accessed without OS intervention.
  - The OS must track which pages are cached and which can/should be evicted.
- Tracking LRU in hardware is expensive/impractical (we saw this with caching; it's even worse here with millions of pages).

Recall the Clock Algorithm

1. Imagine that all your physical pages are arranged around a clock.
2. Each time a page is accessed, the HW sets its use bit to 1.
3. Right now, we have virtual pages 1-11 in memory and their use bits are all 0.



Use bit = 0

Use bit = 1

Page reference stream 1 3 1 1 6 55 4 5 77

How do we handle modified pages?

- We can easily keep track of pages that are modified
  - Add a Dirty bit to each PTE that is set on Write operations.

How do we handle modified pages?

- We can easily keep track of pages that are modified
  - Add a Dirty bit to each PTE that is set on Write operations.
- What if page 7 in the last example had been dirty?

How do we handle modified pages?

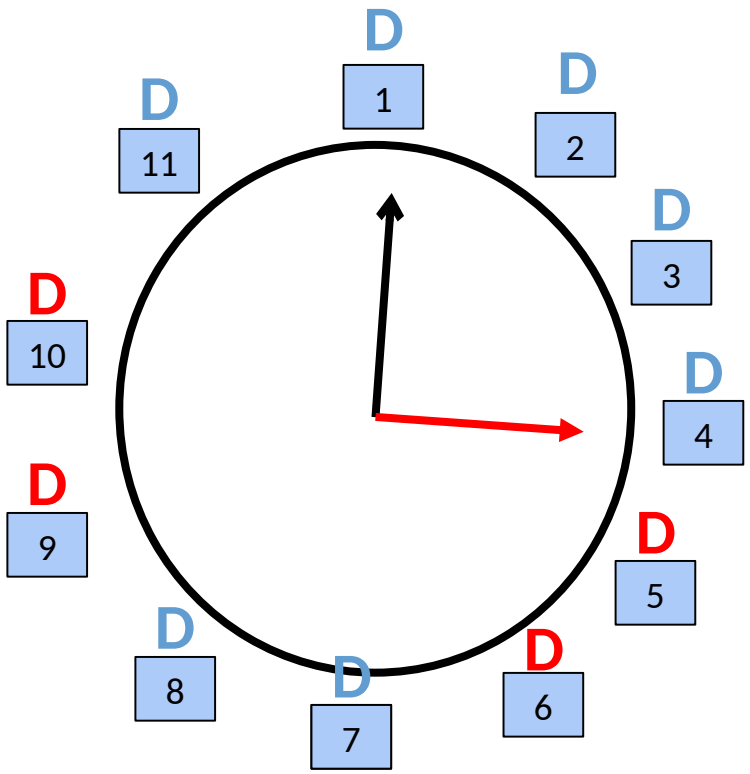
- We can easily keep track of pages that are modified
  - Add a Dirty bit to each PTE that is set on Write operations.
- What if page 7 in the last example had been dirty?
  - We need to write it back to disk before loading page 77.
  - So the replacement takes twice as long.
  - How can we make it faster?

Two-handed clock

1. Hand for replacement

2. Hand for writeback:

- We keep the writeback hand a specific distance ahead of the replacement hand.
- As it moves forward, it sends dirty pages to the disk controller so they can be written back to disk.



Critical point:
We can do this writing while the disk controller is not otherwise occupied, **in parallel** with work happening in the core(s).

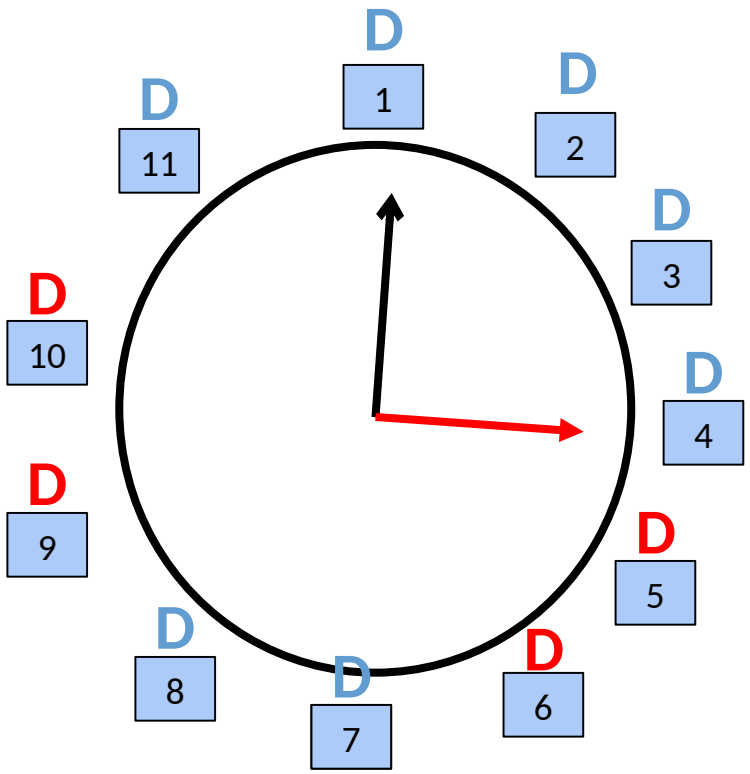
Two-handed clock

Hand for replacement

Hand for writeback

Write Page 3

Read Page 1



Use bit = 0

Use bit = 1

D Dirty bit = 0

D Dirty bit = 1

Two-handed clock

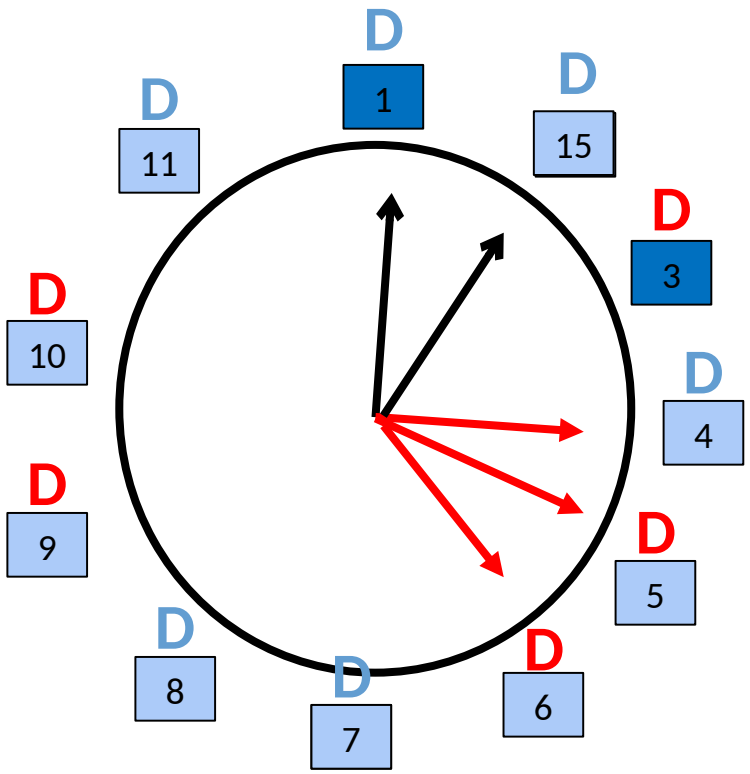
Hand for replacement

Hand for writeback

Write Page 3

Read Page 1

Read Page 15



Use bit = 0

Use bit = 1

D Dirty bit = 0

D Dirty bit = 1

Lots of policy decisions (that we won't discuss!)

- How far ahead of the replacement hand should the write hand be?
- How many pages should we write at a time?
- When should we move the write hand?

Wrapping Up

Virtual memory wraps us up nicely, because it encompasses many topics we've discussed:

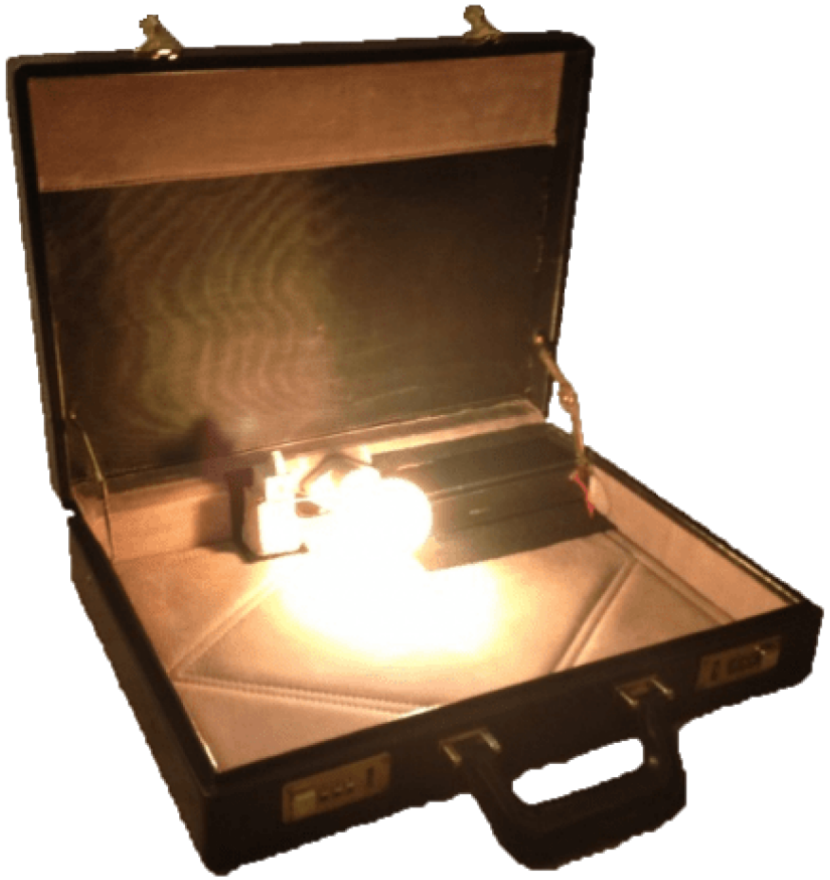
- What hardware does
- Hardware/software interaction
- Managing memory (everything is a cache!)
- Similarities between files and virtual memory
- Management and exploitation of parallelism
- How tradeoffs at multiple levels impact user programs' performance

You can learn about and take advantage of all these elements of your system!

Preparing for the Final

- Get a good night's sleep before your exam session!
- Remind yourself that you can do this. You're an awesome, capable person, and you've worked hard at this!
- And also:
  - Review: in-/pre-class exercises, quizzes, practice quizzes
  - Review tutorials, slides, textbook, labs
  - Form a study group and challenge each other! Adapt existing questions and invent new ones!

Now you know what's
inside the magic briefcase…



Have a great break!

(If "huh?": Check Canvas and then watch Pulp Fiction.
Warning: lots of violence, vulgarity. In Pulp Fiction, not Canvas)