Virtual Memory

Hakim Weatherspoon
CS 3410
Computer Science
Cornell University

[Weatherspoon, Bala, Bracy, McKee, and Sirer]
Where are we now and where are we going?

• How many programs do you run at once?
  • a) 1
  • b) 2
  • c) 3-5
  • d) 6-10
  • e) 11+
Big Picture: Multiple Processes

- Can we execute *more than one* program at a time with our current RISC-V processor?
  - a) Yes, no problem at all
  - b) No, because memory addresses will conflict
  - c) Yes, caches can avoid memory address conflicts
  - d) Yes, our modified Harvard architecture avoids memory address conflicts
  - e) Yes, because we have multiple processors (multiple cores)
Big Picture: Multiple Processes

How to run multiple processes?

• *Time-multiplex* a single CPU core (**multi-tasking**)
  • Web browser, skype, office, … all must co-exist

• Many cores per processor (**multi-core**) or many processors (**multi-processor**)  
  • Multiple programs run *simultaneously*
Processor & Memory

- CPU address/data bus...
- ... routed through caches
- ... to main memory
  - Simple, fast, but...
Multiple Processes

• Q: What happens when another program is executed concurrently on another processor?

• A: The addresses will conflict
  • Even though, CPUs may take turns using memory bus
Multiple Processes

• Q: Can we relocate second program?
Solution? Multiple processes/processors

• Q: Can we relocate second program?
• A: Yes, but…
  • What if they don’t fit?
  • What if not contiguous?
  • Need to recompile/relink?
  • …
Big Picture: (Virtual) Memory

Give each process an illusion that it has exclusive access to entire main memory.
But In Reality…

Process 1

Process 2

Physical Memory
How do we create the illusion?

Process 1

A
B
C
D

Process 2

E
F
G
H

Physical Memory
How do we create the illusion?

All problems in computer science can be solved by another level of indirection.

– David Wheeler
How do we create the illusion?

Virtual Memory
(just a concept; does not exist physically)
How do we create the illusion?

Process 1 wants to access data C
Process 1 thinks it is stored at addr 1
So CPU generates addr 1
This addr is intercepted by MMU
MMU looks at the mapping
Virtual addr 1 -> Physical addr 9
Data at Physical addr 9 is sent to CPU
And that data is indeed C!!!

Virtual Memory (just a concept; does not exist physically)

Physical Memory
How do we create the illusion?

Process 1

Virtual Memory

Disk

Physical Memory

Map virtual address to physical address

Memory management unit (MMU) takes care of the mapping
Big Picture: (Virtual) Memory

• From a process’s perspective –
  - Process only sees the virtual memory
    ✓ Contiguous memory
Big Picture: (Virtual) Memory

• From a process’s perspective –

  Process only sees the virtual memory
  ✓ Contiguous memory
  ✓ No need to recompile - only mappings need to be updated
Big Picture: (Virtual) Memory

• From a process’s perspective –
  - Process only sees the virtual memory
    ✓ Contiguous memory
    ✓ No need to recompile - only mappings need to be updated
Big Picture: (Virtual) Memory

• From a process’s perspective –

  - Process only sees the virtual memory
    - Contiguous memory
    - No need to recompile - only mappings need to be updated
    - When run out of memory, MMU maps data on disk in a transparent manner
Next Goal

• How does Virtual Memory work?

• i.e. How do we create the “map” that maps a virtual address generated by the CPU to a physical address used by main memory?
Next Goal (after spring break!)

• How does Virtual Memory work?

• i.e. How do we create the “map” that maps a virtual address generated by the CPU to a physical address used by main memory?
Virtual Memory Agenda

What is Virtual Memory?
How does Virtual memory Work?

- Address Translation
- Overhead
- Paging
- Performance
Have a great Spring Break!!!