Operating Systems!

An Overview of Computer Systems (Part II)

Prof. Travis Peters

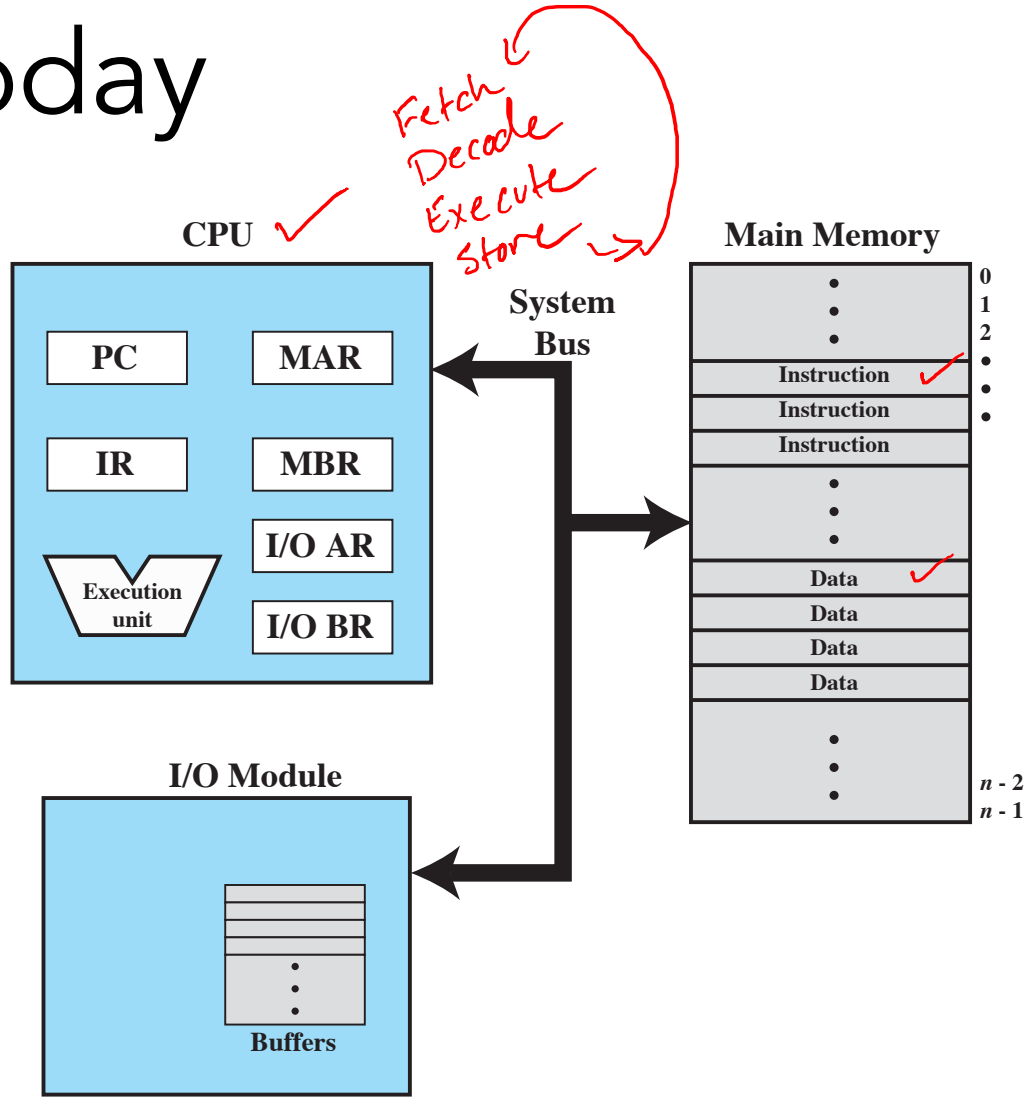
Montana State University

CS 460 - Operating Systems

Fall 2020

https://www.cs.montana.edu/cs460

Today



Announcements

- ***Deadlines:***
  Sundays @ 11:59 PM (MST)

Learning Objectives

- Review basic elements of **computer systems** and their interrelationships
- Review important system concepts *(e.g., ~~processor execution~~, ~~interrupts~~, **memory**, I/O)*

Memory

What follows are the big ideas,
which gives us context for the role and purpose of the OS.

Memory

- Design constraints on a computer's memory
  - How much?
  - How fast?
  - How expensive?

- Realities
  - Want ***fast access*** to ***as much memory as possible!***
  - If the memory is available,
  - applications will likely be developed to use it
  - The memory bottleneck
    (reads/writes are slowwww compared to processor)
  - The ***cost*** of memory can't be ignored

"640K ought to be enough for anybody."
- Bill Gates, 1981 (?)

Designing System Memory

Breakout Activity!

Given your budget, and the cost/capacity/access times, how would you build system memory?

Budget = $1000

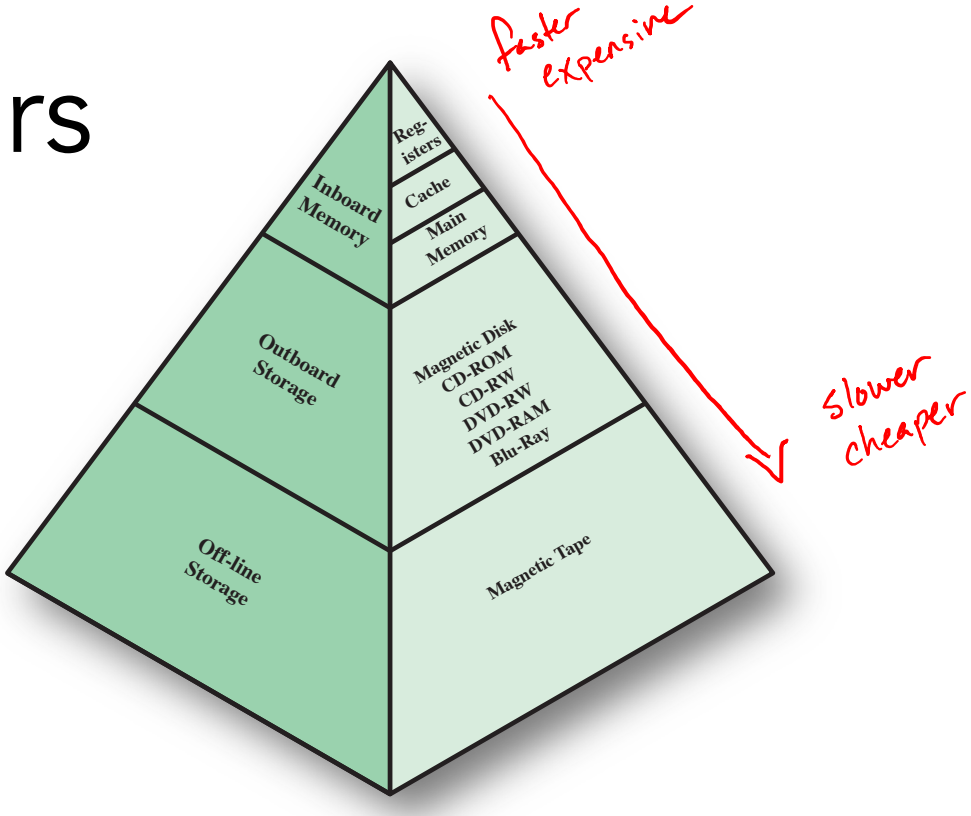
- 1KB memory, 0.1μs access time, $500
- 1MB memory, 1μs access time, $100
- 1GB memory, 1ms access time, $10

0μs? :'(

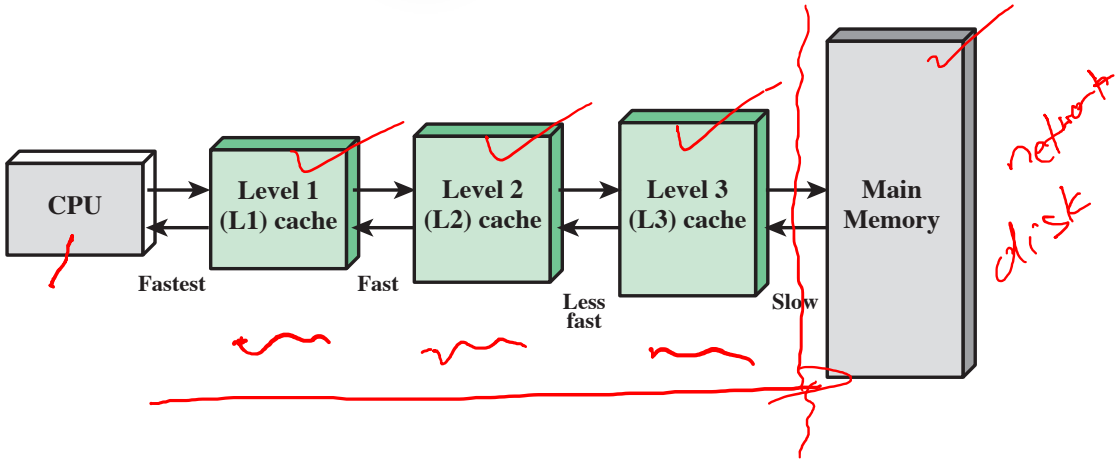
① split ✓✓
(maybe 1KB not worth it)

Memory & Practical Matters

- The Memory Hierarchy
  - want *fast access* to a *large amount of memory*
  - each type of memory offers its own trade-offs
  - *IMPORTANT IDEA: locality of reference:* memory access tends to cluster (e.g., loops, subroutines)
    ➔ cache, virtual memory, etc.



- Cache Memory
  - keep as much memory *as close as possible*...
    ...but still *support large amounts of memory.*
  - Multiple levels of cache... HW cache is a more advanced topic beyond the scope of this class (cache design, *cache coherency, etc.)*



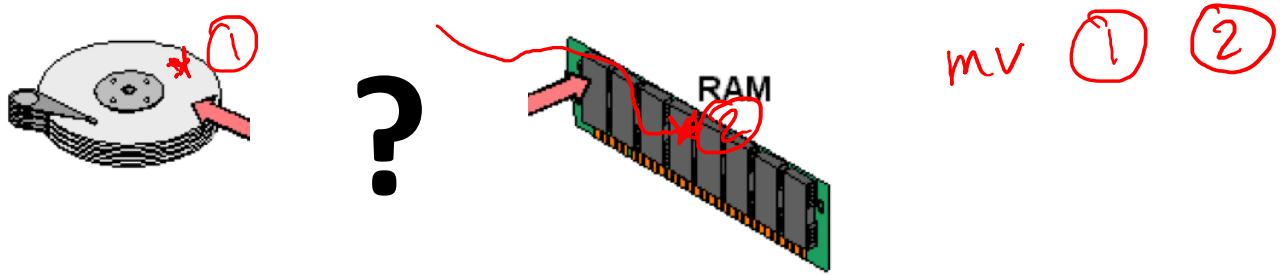
Memory & Input/Output (I/O)

What follows are the big ideas,
which gives us context for the role and purpose of the OS.

Designing System I/O

Breakout Activity!

(think like a computer!)



(1) You want some data. How do you get it?

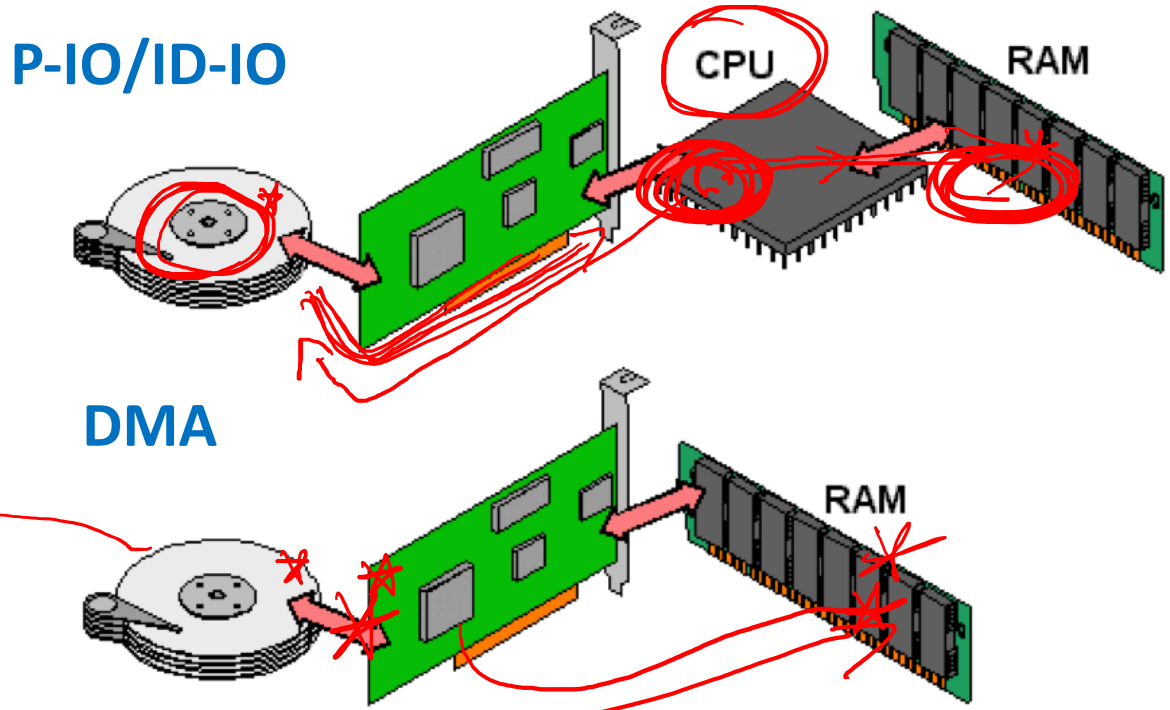
(2) ...Your data is ready!

(But how do you know? How do you get it?)

3 Techniques for I/O & Memory Access

- **Programmed I/O** (*active*; processor polls I/O device)
- **Interrupt Driven I/O** (*assist*; I/O device does work; interrupt processor for help)
- **DMA** (*delegate*; DMA module give OP, DEV, ADDR, #WORDS; send interrupt when done)

Direct Memory Access



Real Computers

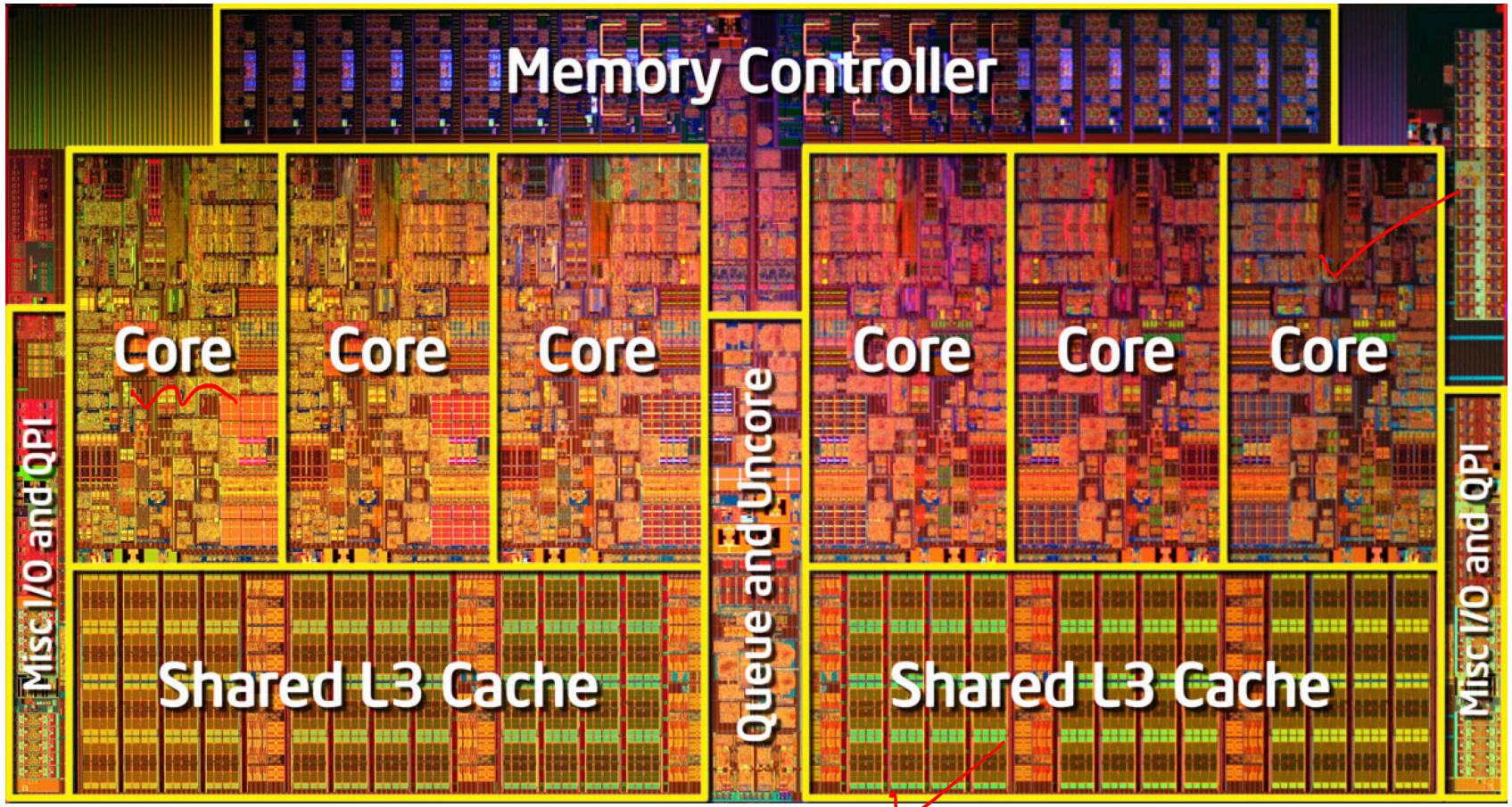
Yes, all of this stuff we are talking about is real!

What follows are the big ideas,
which gives us context for the role and purpose of the OS.

Multicore Systems

Access to RAM, disk, and other I/O via Memory Controller and I/O controllers

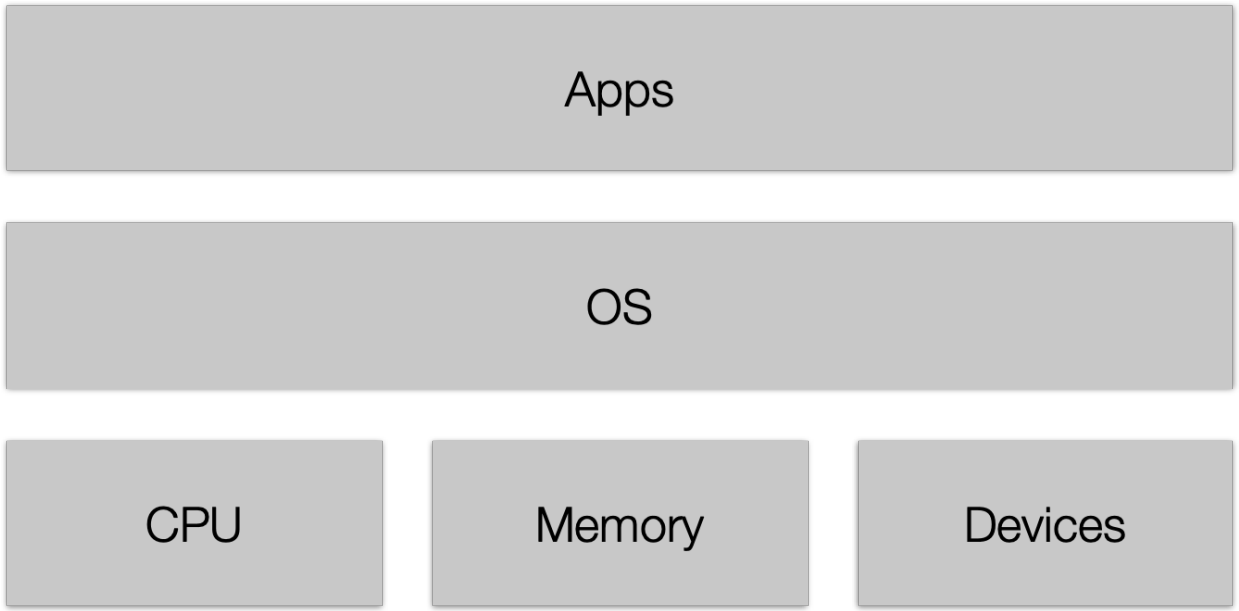
Private cache (L1/L2) within cores



L3 cache shared across multiple cores

http://www.gizmodo.com.au/2010/03/intels-6-core-gulftown-gets-tested-blows-us-away/

Summary: The Basics of Computer Systems



Questions you can answer....

- *Why do we need a CPU?*
- *How does a CPU work?*
- *How do we utilize a CPU effectively?*
- *Why do we need (different types of) memory?*
- *Why do we need I/O devices?*
- *How do we utilize I/O devices efficiently?*
- *Why do we need a system bus?*
- *Why do we need an OS?*

Minute Paper

Please spend 1 minute (or a few) jotting down thoughts in response to the following questions:

- *What is one thing (or a few) that is now much clearer after this review?*
- *What is one thing (or a few) that is still unclear after this review?*

Please respond via this Google Form!

https://forms.gle/dxRexo9t9CBbbFm38