CSCI 2410 Systems Programming

Fall 2026

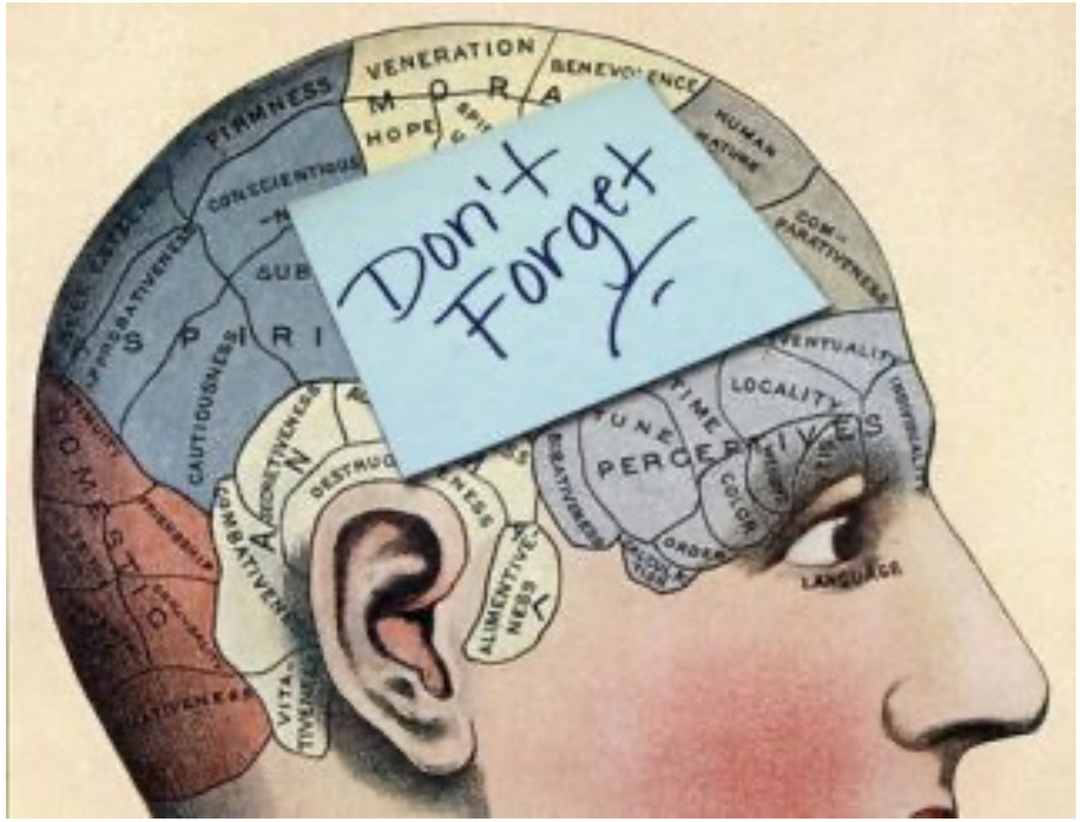
Instructor: Jie Zhou

Department of Computer Science

George Washington University

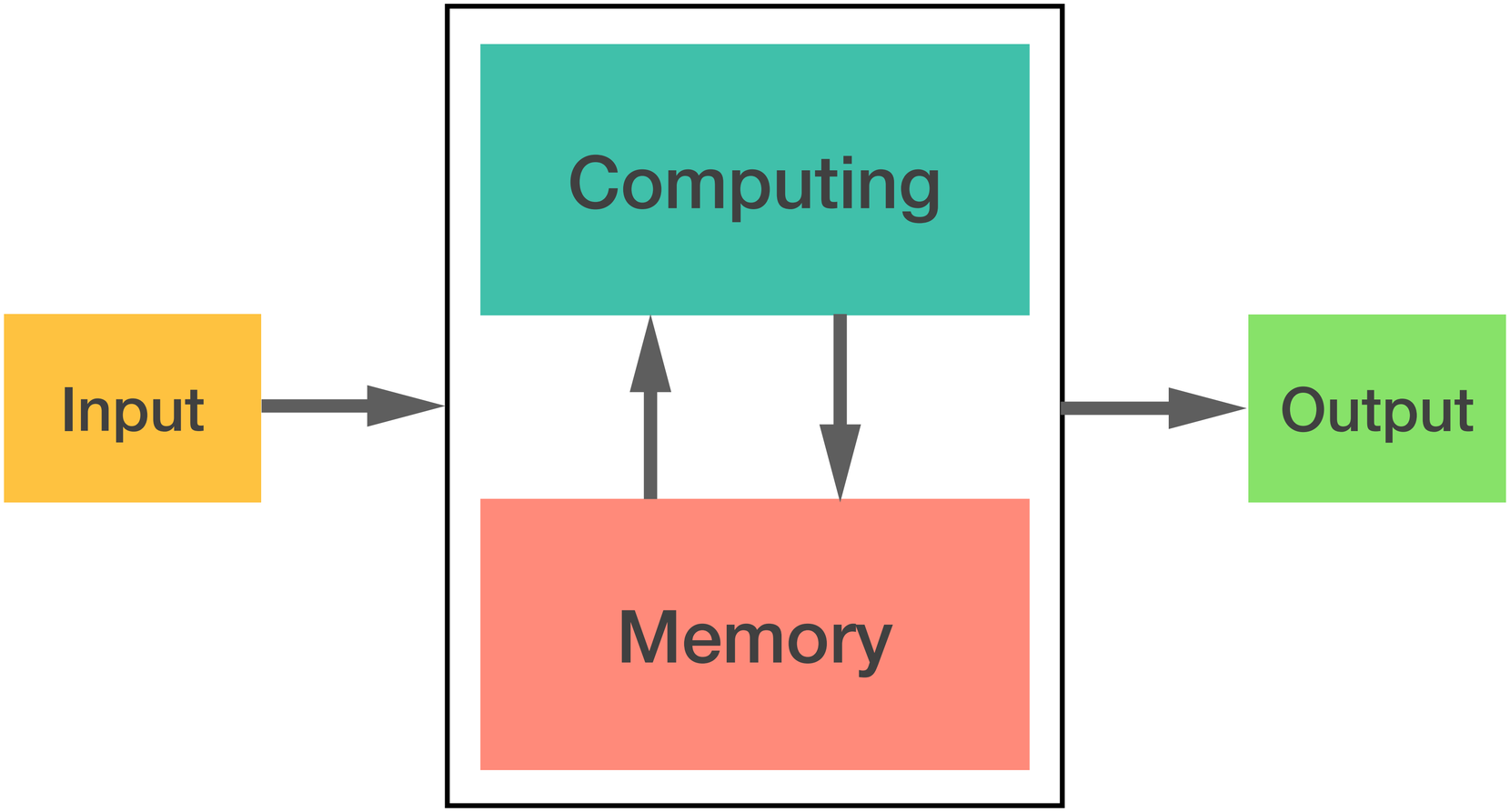
Slides materials are mostly credited to Sibin Mohan and Gabriel Parmer at GWU.

Memory


VENERATION
FIRMNESS
HOPE
CONSCIENTIOUS
HUMAN NATURE
Don't Forget
LOCALITY
LANGUAGE
ALIMENTIVENESS

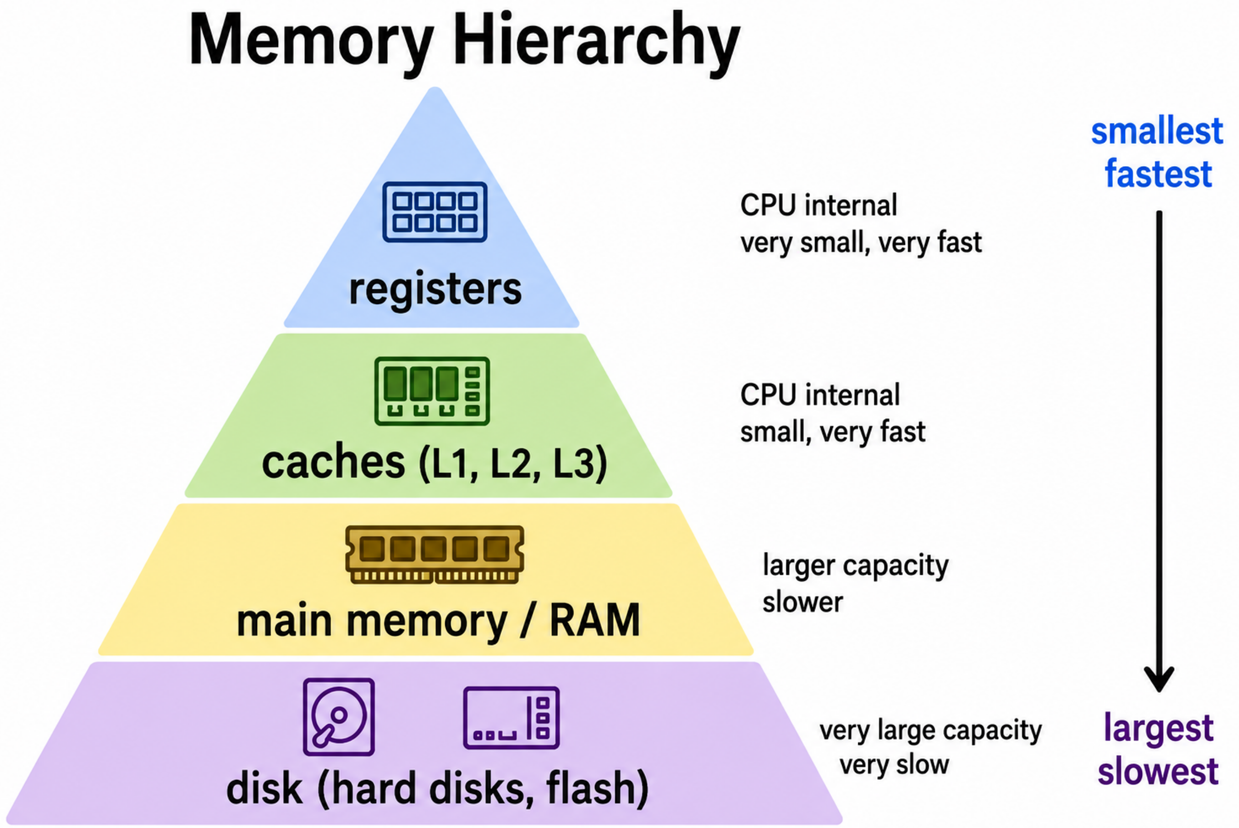
Why do computers need memory?

Architecture of Modern Computers



Computer Memory

- Different types, sizes, and speeds
  - registers
  - caches (L1, L2, L3)
  - main memory / RAM
  - disk (hard disks, flashes, etc.)



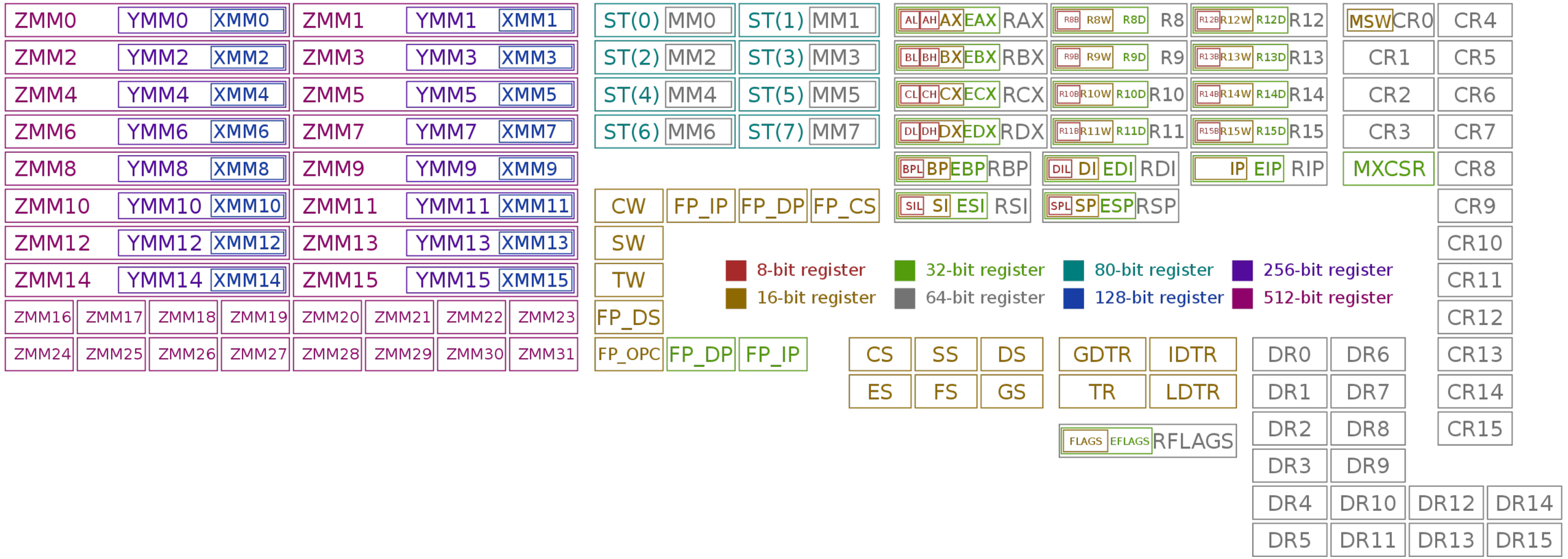
Registers

- Fastest form of memory
- Part of CPUs; what CPU uses for computing
- Each CPU has a set of registers.
- e.g., `a = b + 3;`
  - b is loaded into a register
  - a is computed and the result stored in another register or the same register used by b, if b is not used anymore.

Register Width

- Defined by CPU
- “32-bit processor” -> 32-bit registers
- “64-bit processor” -> 64-bit registers

All (Known) x86 Registers for the 64-bit Architecture

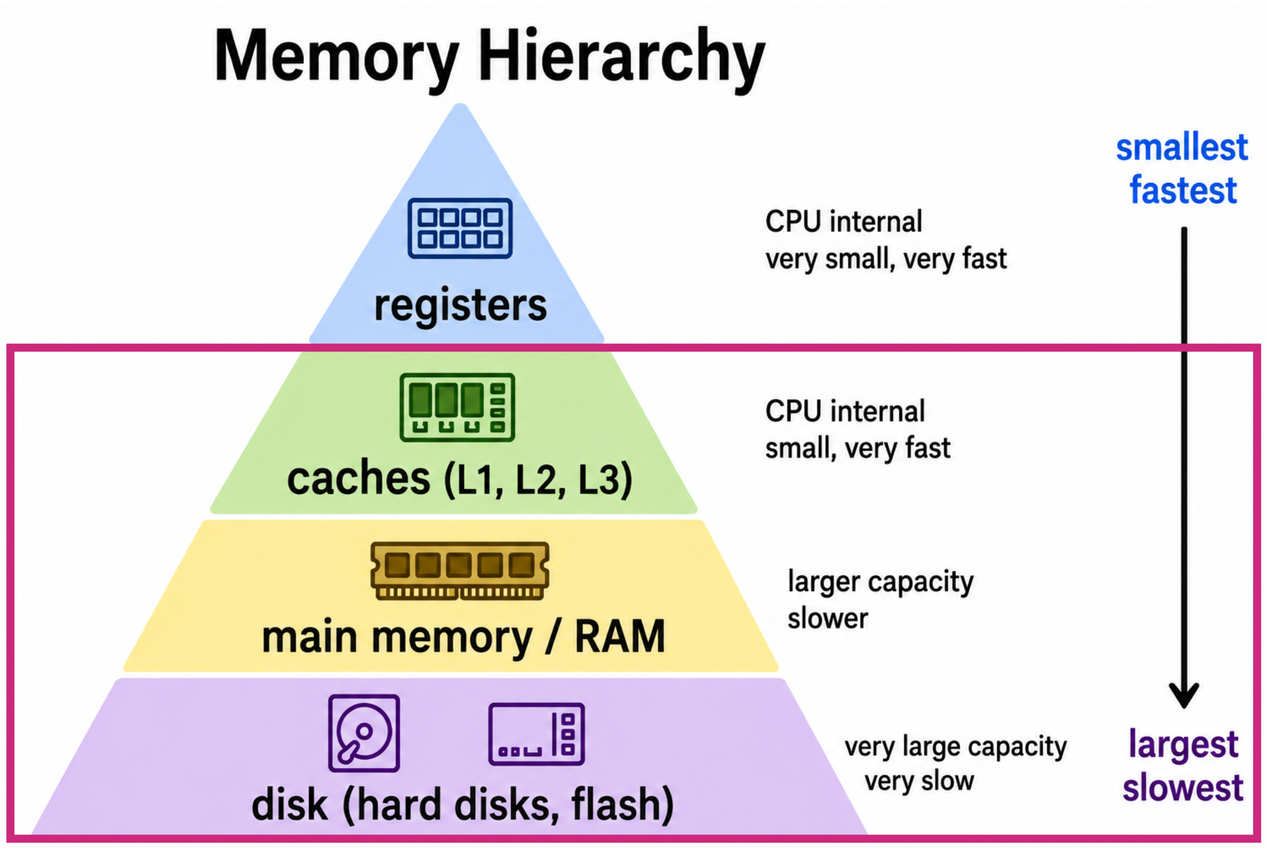


Why Do Computers Need Memory?

- Computers process data, and data needs space.
  - Data type: code and data to be processed by code
    - e.g., Compilers treat code as a type of data.

Component	Typical Sizes
programs/code	kilobytes (KB) or megabytes (MB)
temporary/runtime values	kilobytes (KB) or megabytes (MB)
persistent data	kilobytes (KB) or terabytes (TB)

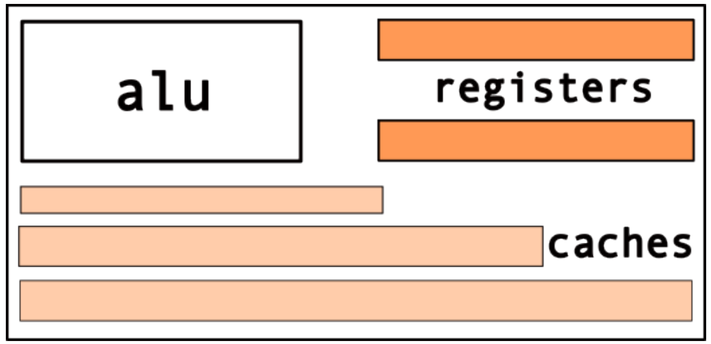
Memory Hierarchy



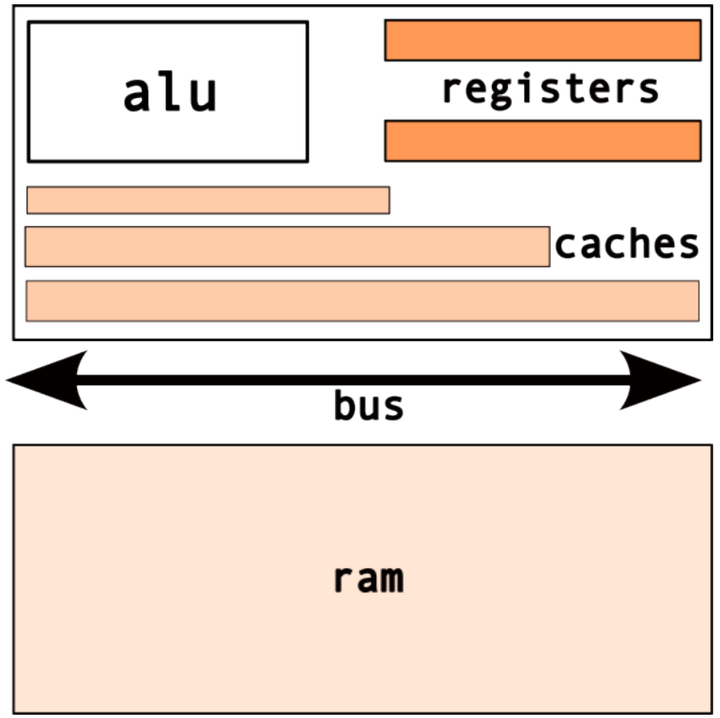
Why do we need so many layers?

- Registers are expensive!
- So are caches.
- Also, registers and caches are volatile.
- Speed vs. size: The faster the memory, the more expensive.

Memory Layout



Memory Layout



Caches

- Fast, on-chip, volatile memory
- (Largely) Transparent to programs and programmers
- Used by CPU to improve performance
- Typical sizes, hundreds of KBs to hundreds of MBs

Main Memory

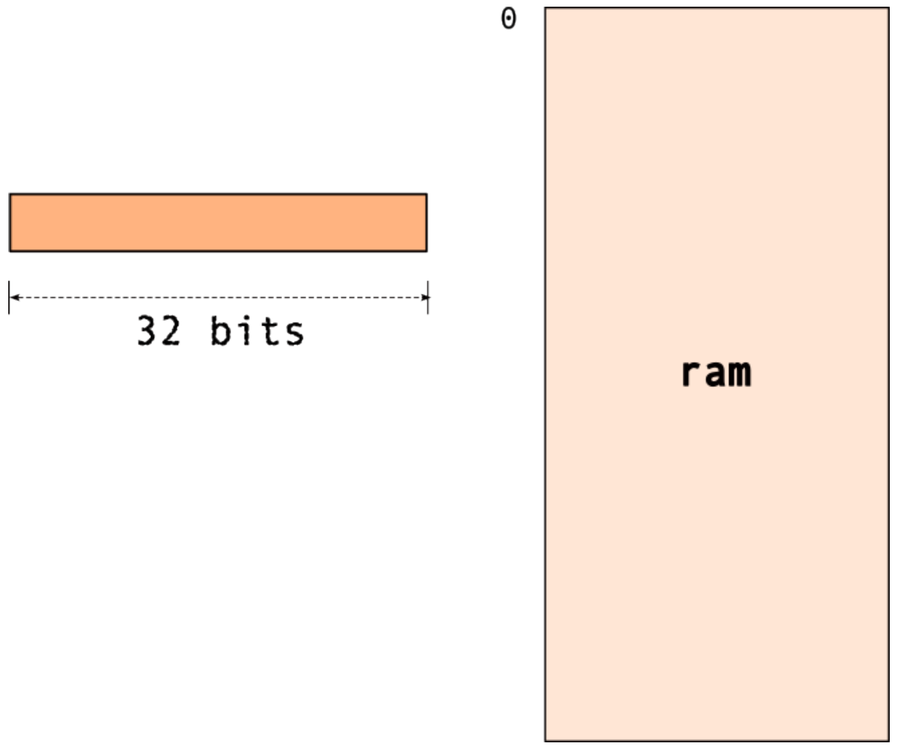
- Random Access Memory (RAM), also known as DRAM
- Typical sizes, GB range
- Limited by “width” of CPU
  - e.g., 32-bit CPU —> 4 GB of memory (generally)

Register Width, Memory Size, ….

Main Memory

- Random Access Memory (RAM), also known as DRAM
- Typical sizes, **GB range**
- Limited by “width” of CPU
  - e.g., 32-bit CPU —> 4 GB of memory (generally)
- Memory is divided into many units, and each unit has its *address*.
- In mainstream systems, memory is *byte-addressable.*

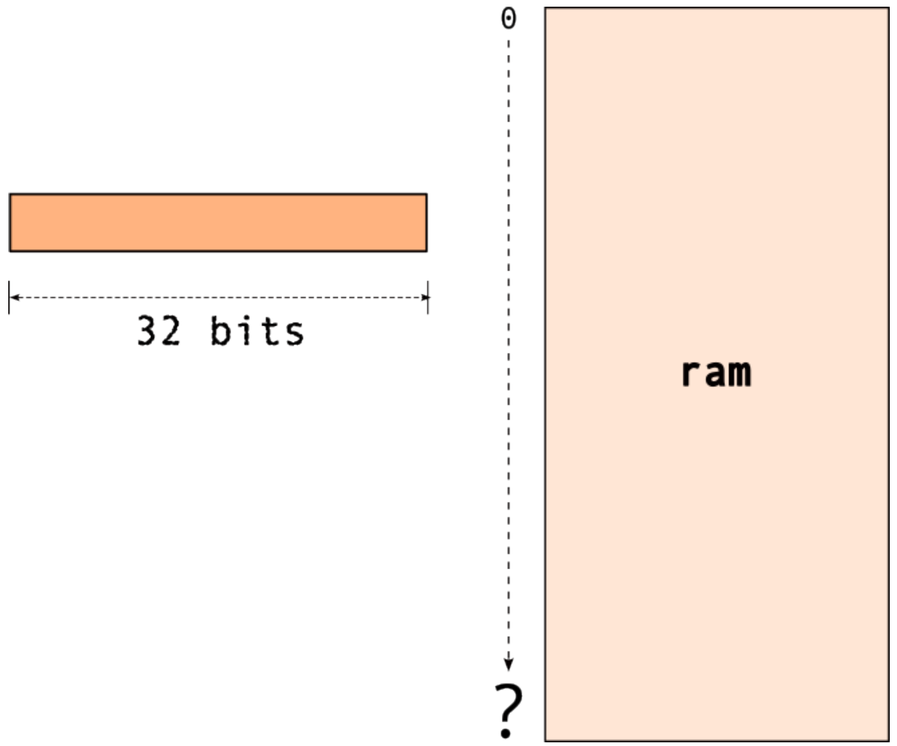
Memory Addressing



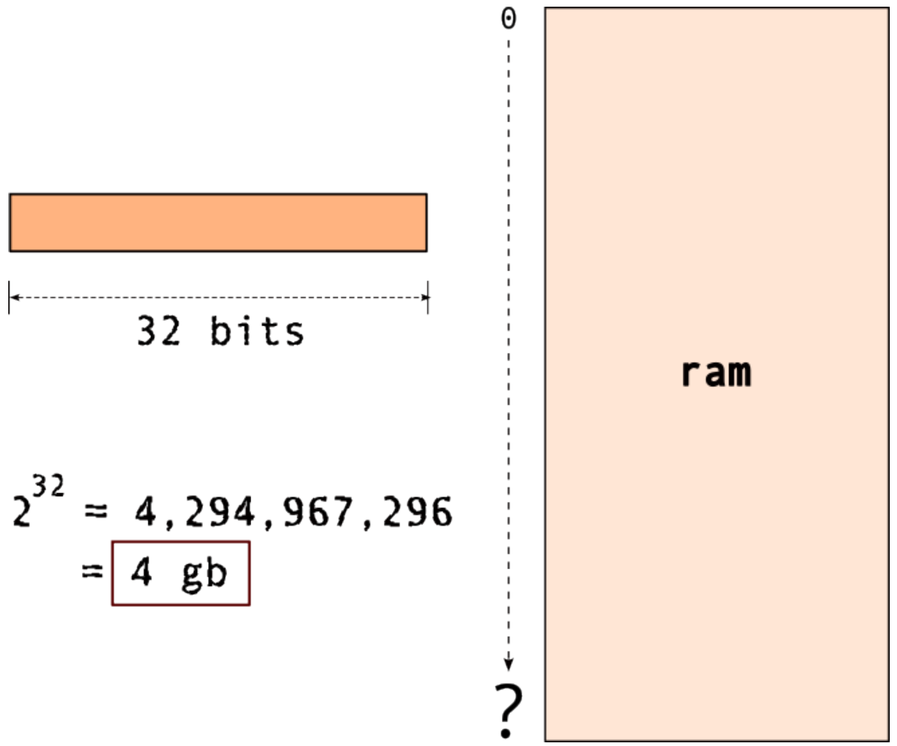
Registers

- Fastest form of memory
- Part of CPUs; what CPU uses for computing
- Each CPU has a set of registers.
- e.g., `a = b + 3;`
  - b is loaded into a register
  - a is computed and the result stored in another register or the same register used by b, if b is not used anymore.

Memory Addressing



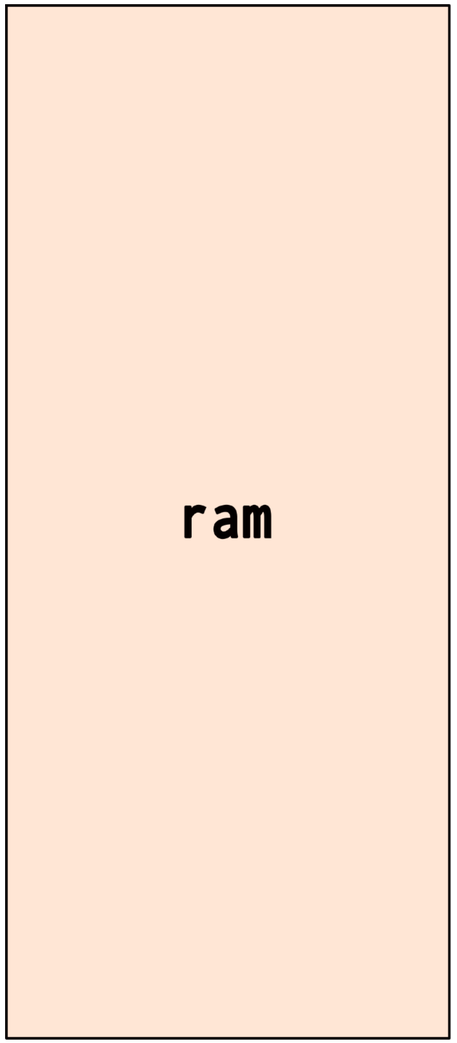
Memory Addressing



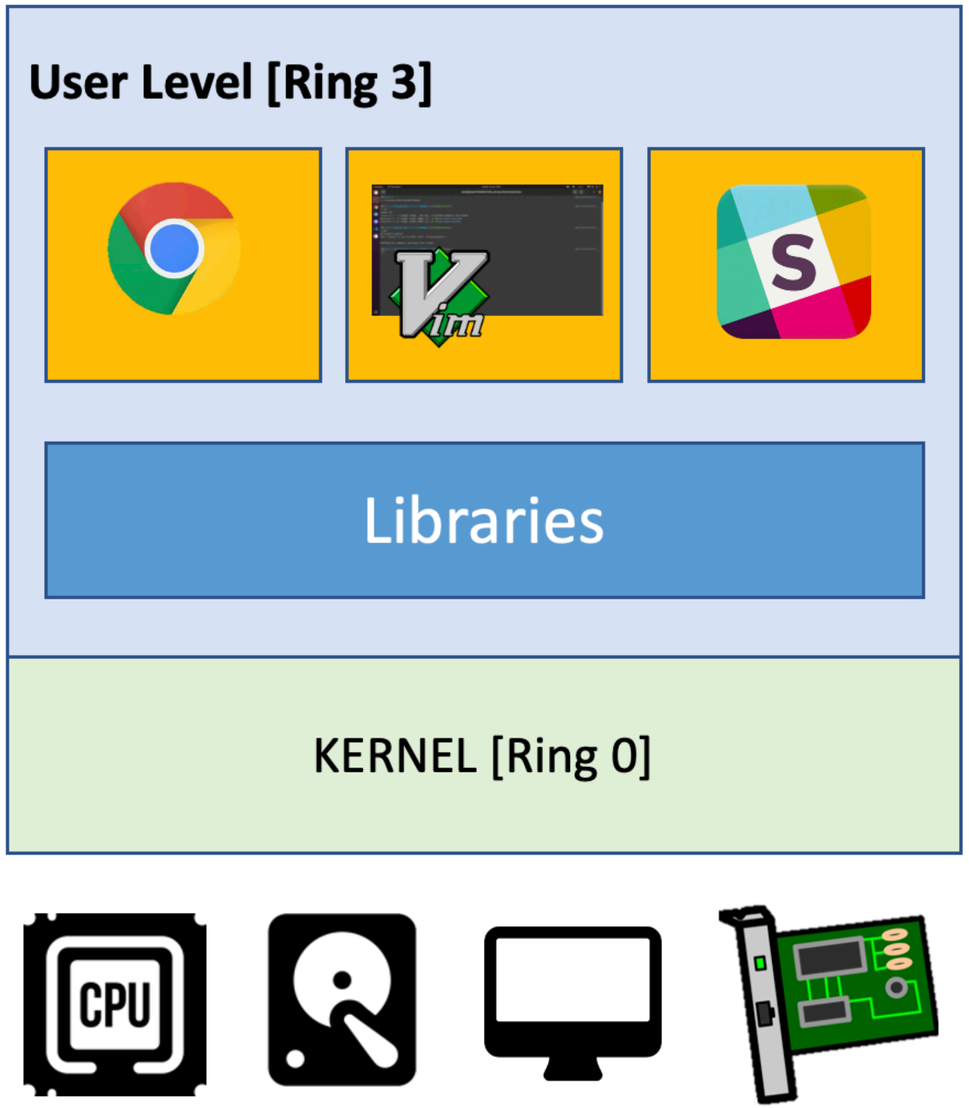
2^{32} = 4,294,967,296

= 4 gb

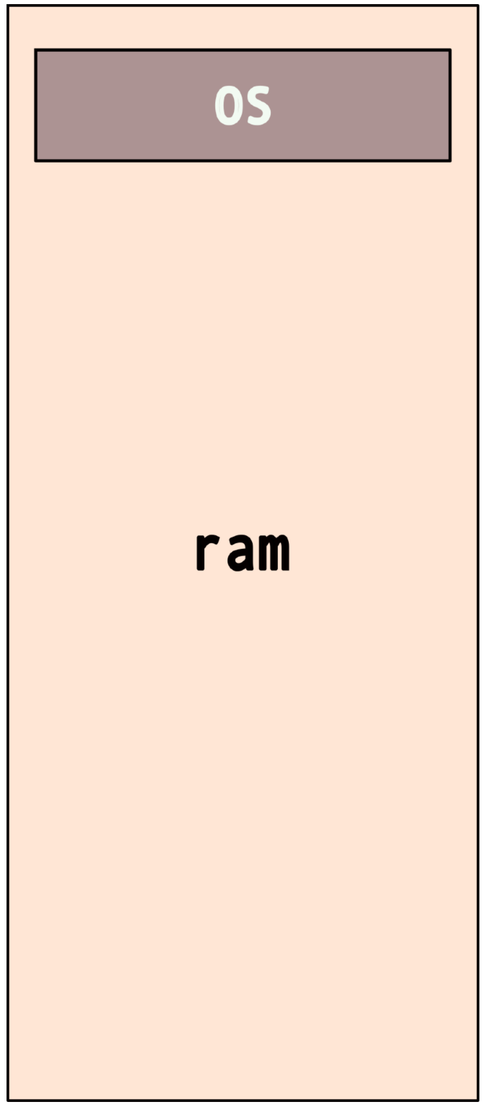
What Goes Into Memory/RAM?



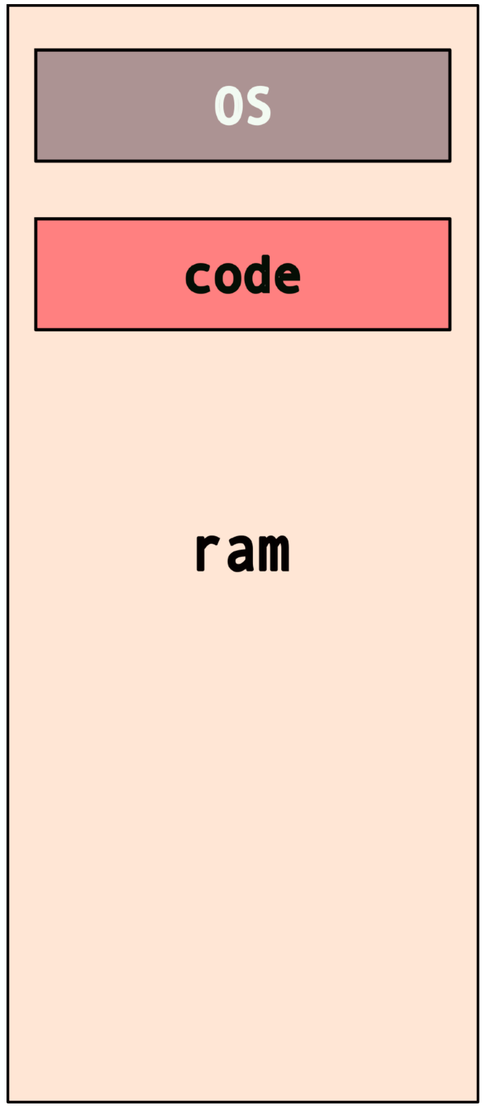
What Goes Into Memory/RAM?



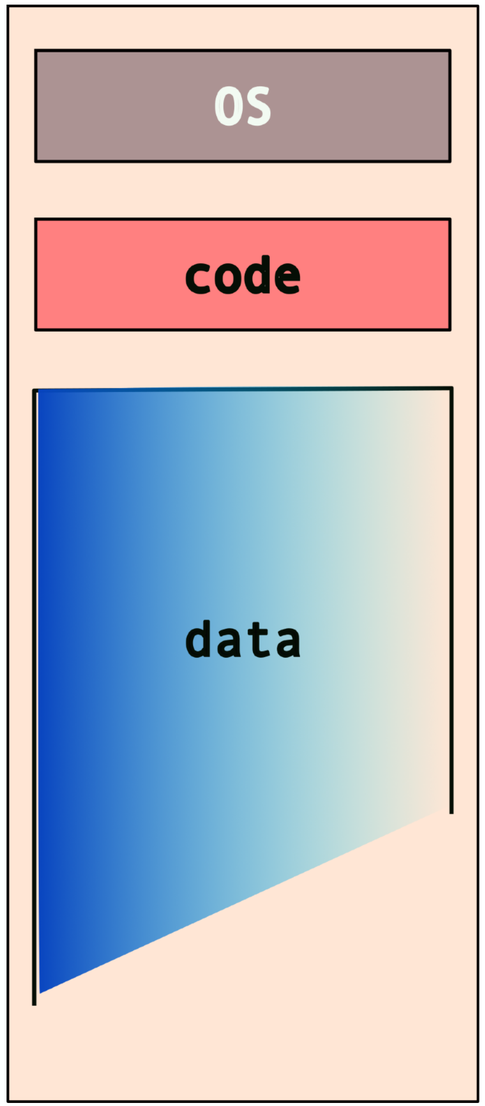
What Goes Into Memory/RAM?



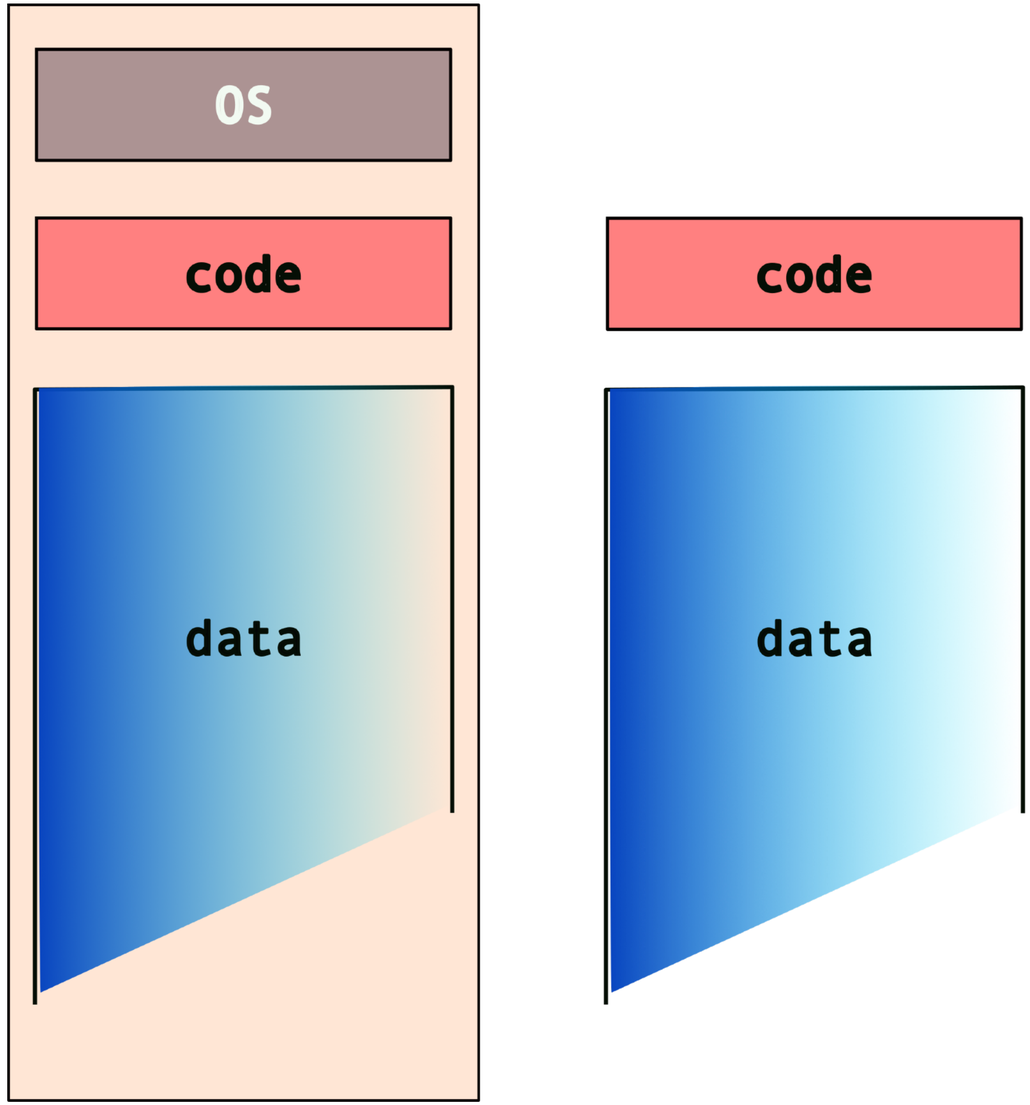
What Goes Into Memory/RAM?



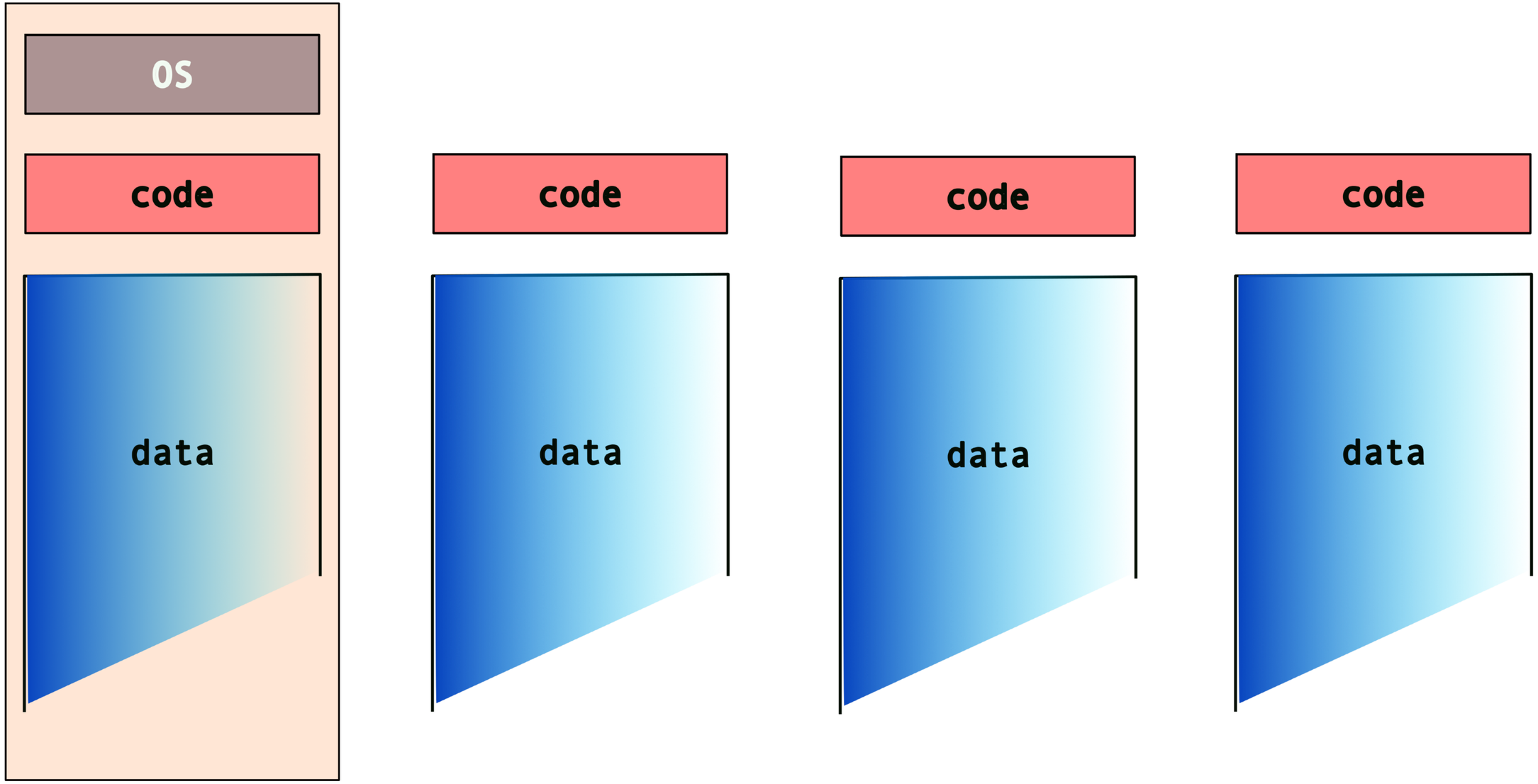
What Goes Into Memory/RAM?



What Goes Into Memory/RAM?



What Goes Into Memory/RAM?



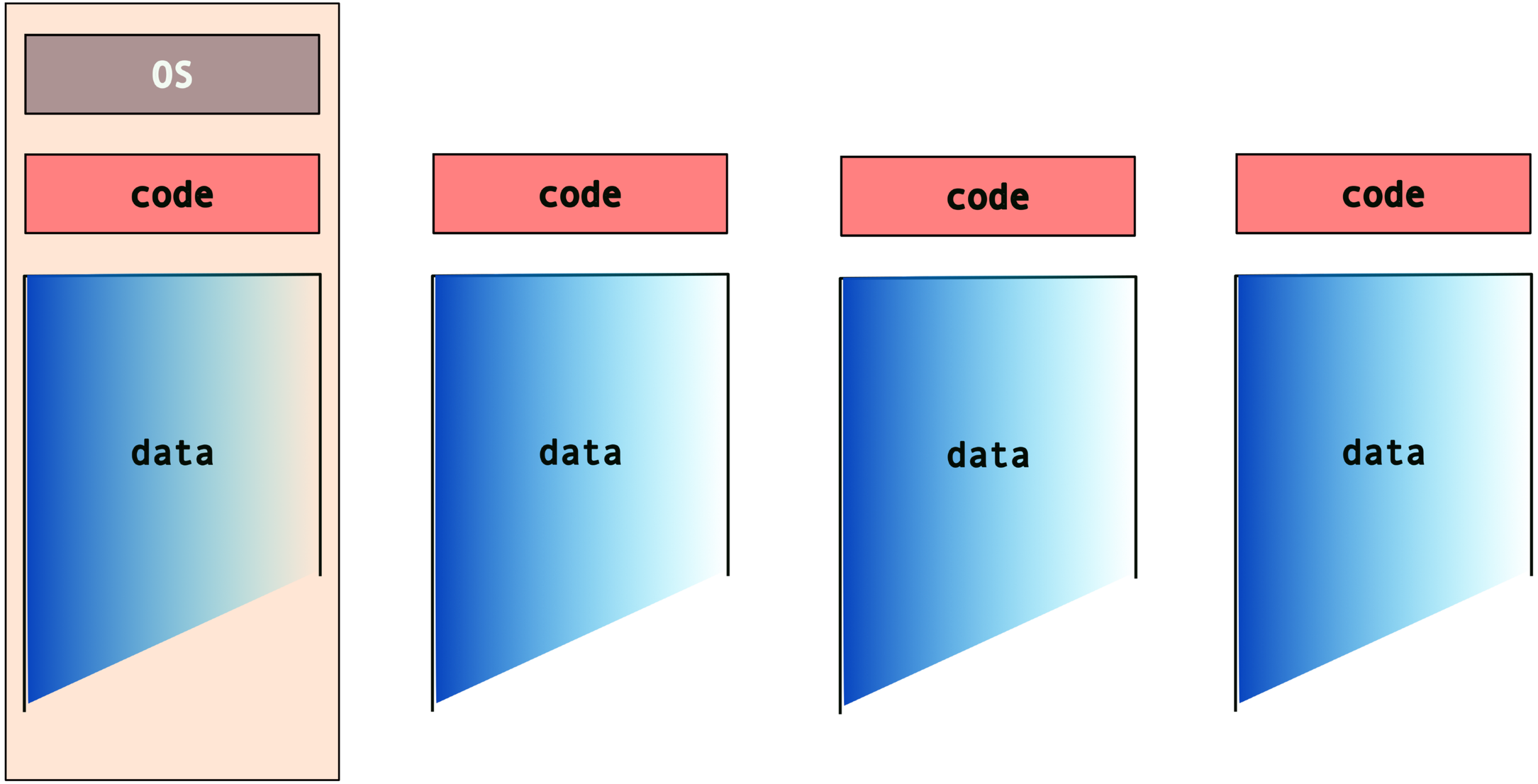
It can quickly get messy to manage all instances in one address space.

Virtual Memory

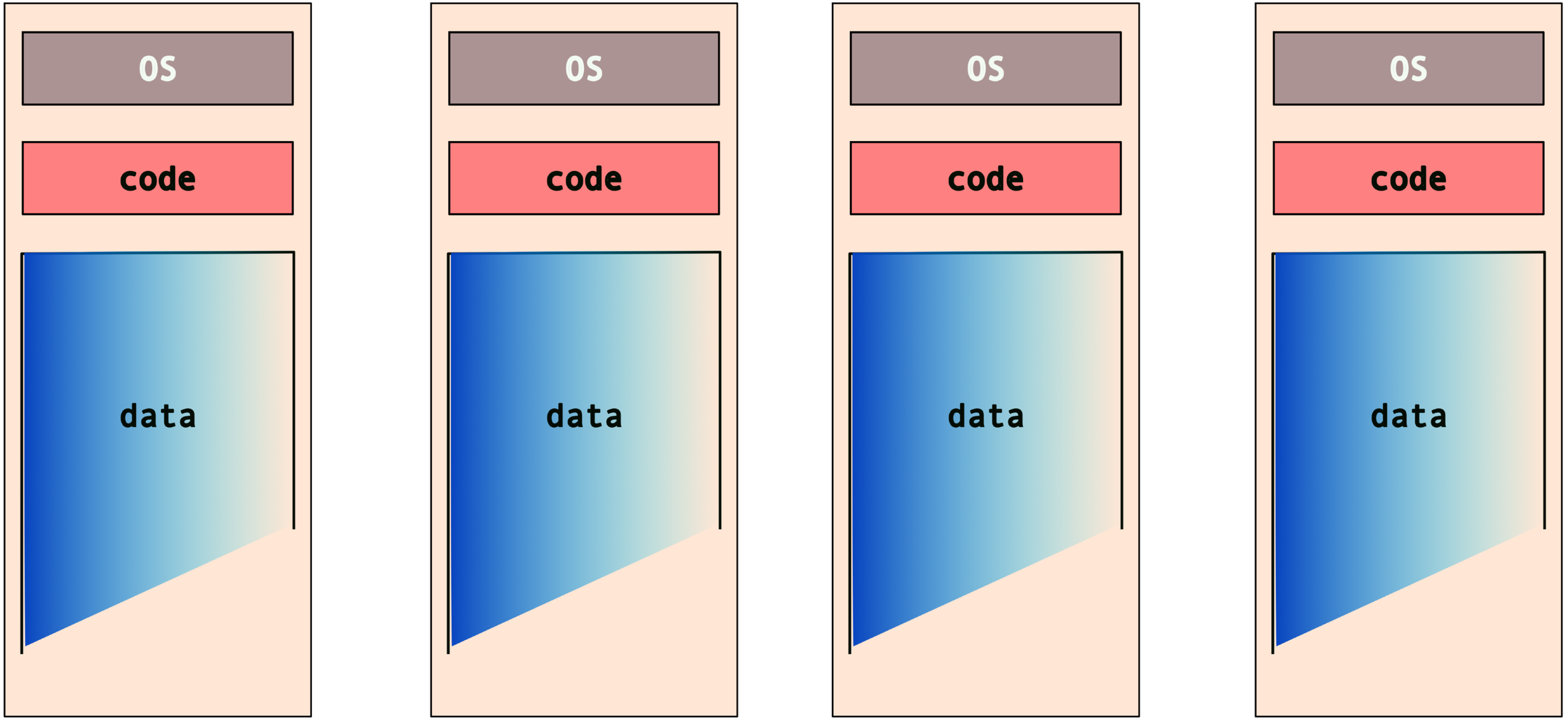
Virtual Memory

- Makes each program “believe” it as the *entire* memory it itself!
- Combination of OS + hardware

So, instead of this:



we get…



Or, at least the programs “think” so!

Virtual Memory

- Makes each program “believe” it as the *entire* memory it itself!
- Combination of OS + hardware
- Programmers need **not** worry about other programs.
- Easier for OS to manage all programs.
- In this course—and in most of your career as a systems programmer—you will mainly deal with virtual memory.