Announcements

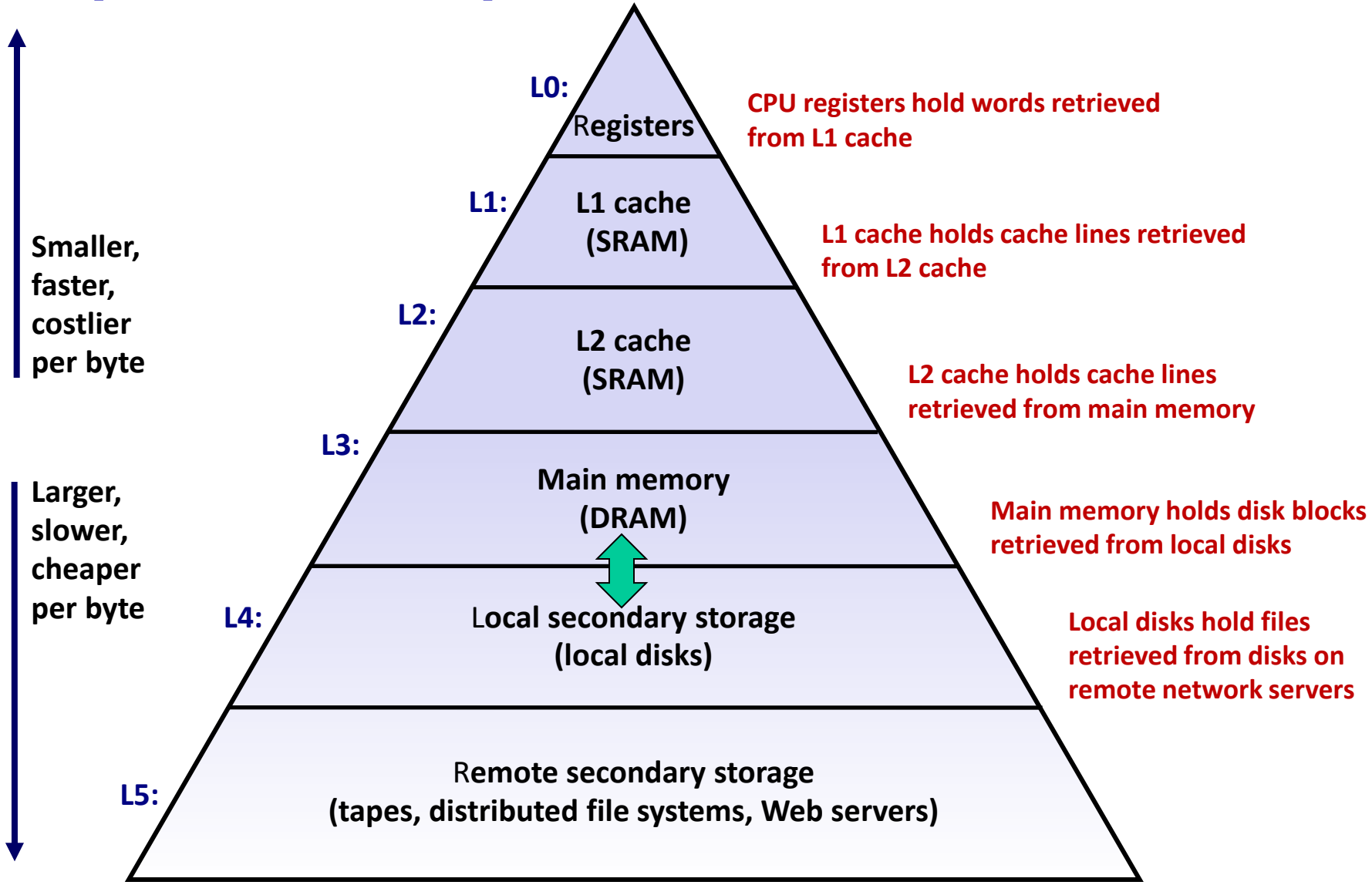
- Assignment 7
  - Due 5 p.m. tomorrow
- Exam 3
  - Friday, November 21
  - Covers up to Cache: Lectures 19-28
  - Format: multiple-choice (30%) / short-answer (70%)

Lecture 29

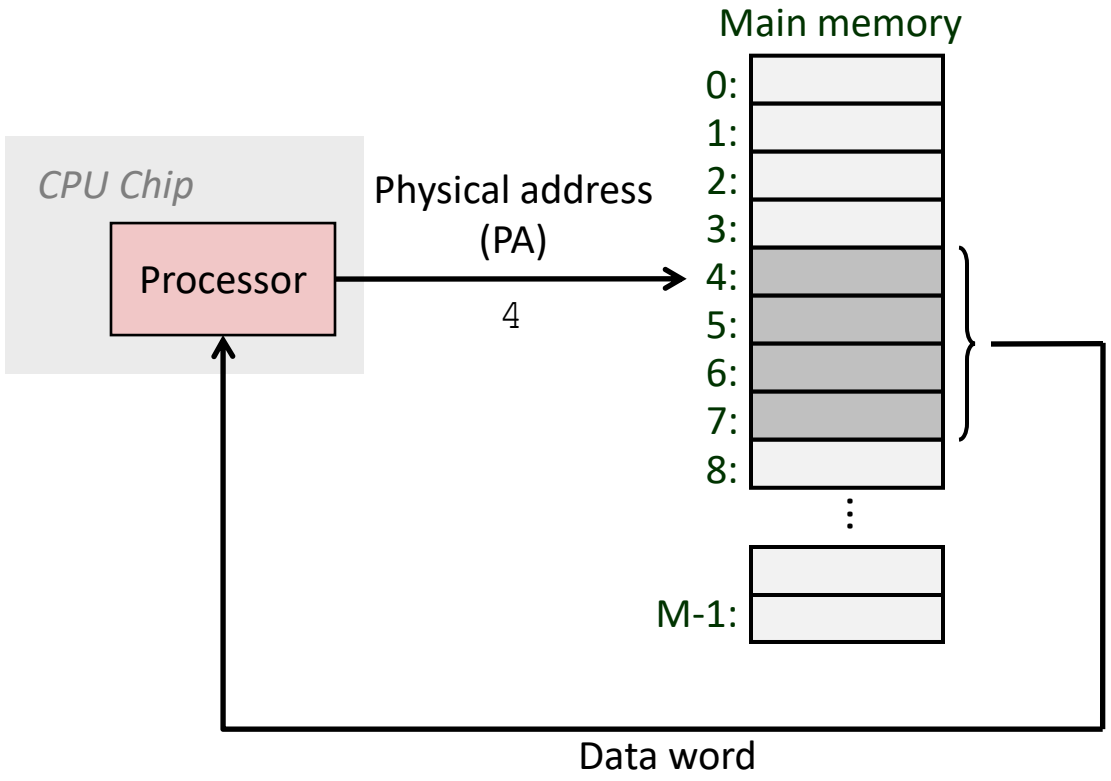
Virtual Memory

CPSC 275
Introduction to Computer Systems

Memory Hierarchy

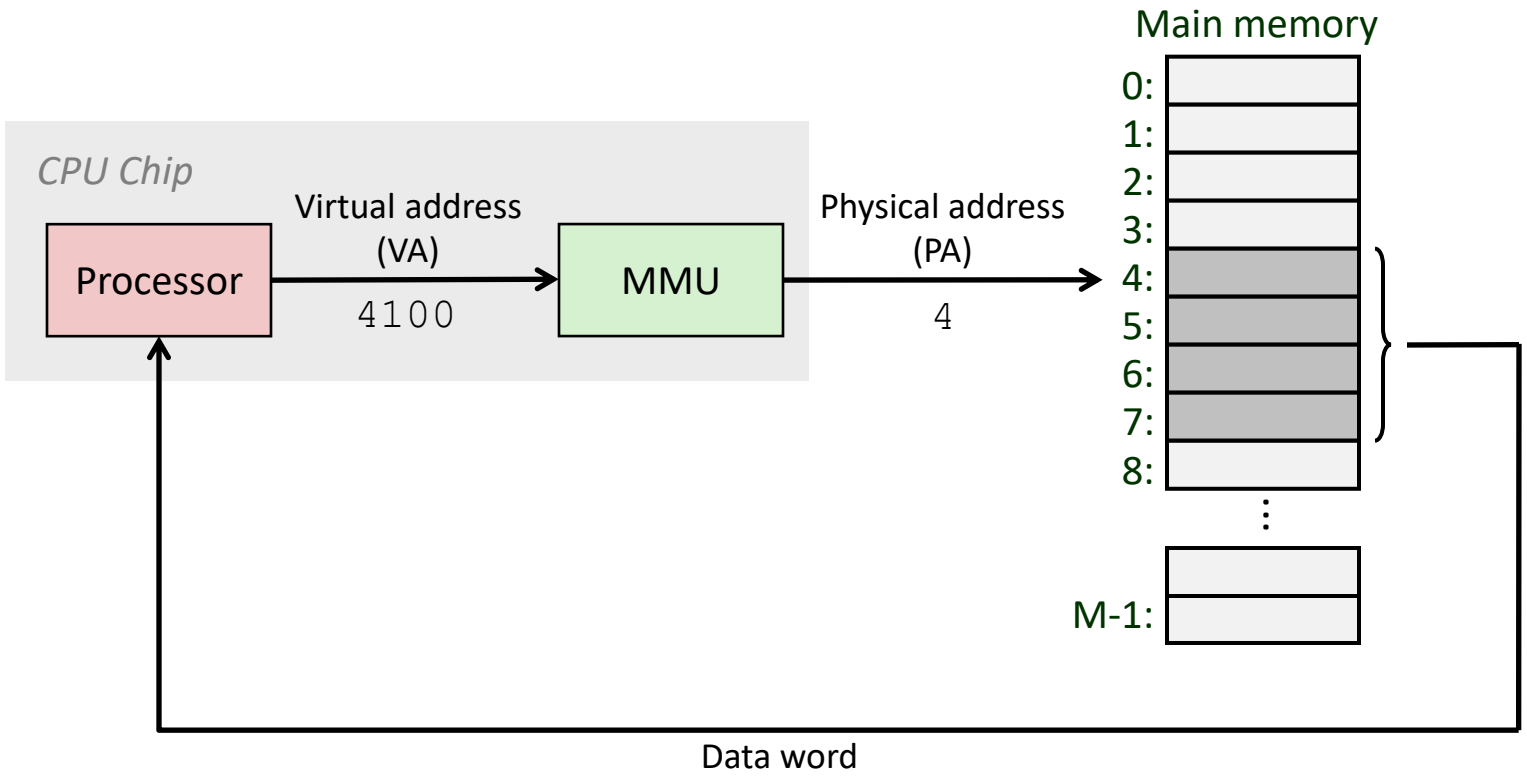


A System Using Physical Addressing



- Used in "simple" systems like embedded systems, e.g., cars, elevators, and digital picture frames

A System Using Virtual Addressing



- Used in virtually all modern computer systems.

Address Spaces

- Virtual address space: set of $N = 2^n$ virtual addresses

  $\{0, 1, 2, 3, \ldots, N\text{-}1\}$

  - Compiler generates *relocatable* virtual addresses

- Physical address space: set of $M = 2^m$ physical addresses

  $\{0, 1, 2, 3, \ldots, M\text{-}1\}$

  - These are actual addresses in DRAM.

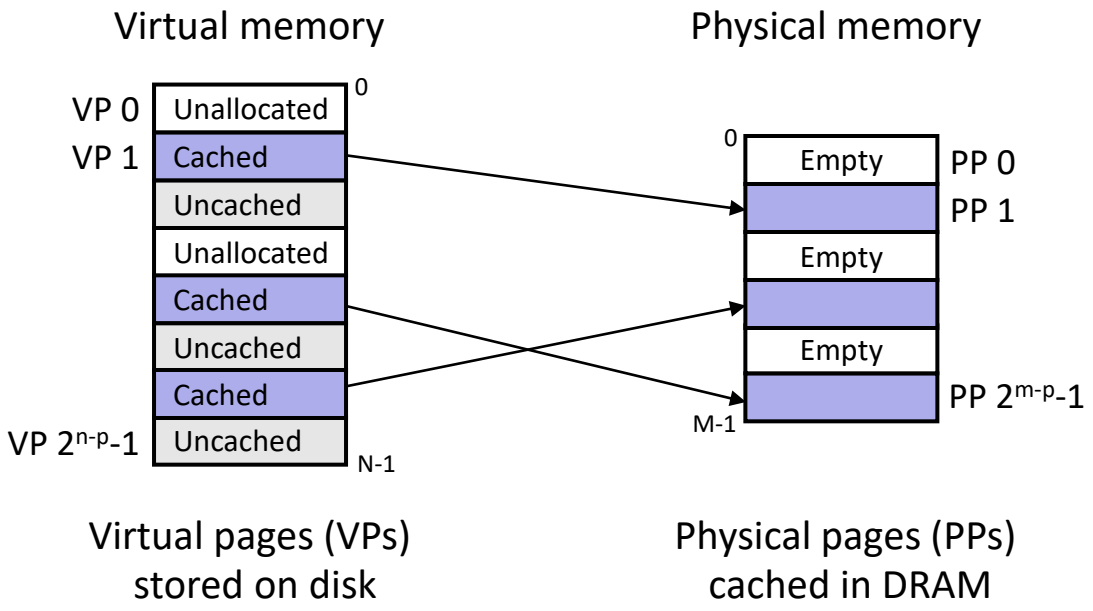
- Every byte in main memory:
  - one physical address
  - one (or more) virtual addresses

Why Virtual Memory?

- Uses main memory efficiently
  - Use DRAM as a cache for the parts of a virtual address space
- Simplifies memory management
  - Each process gets the same uniform linear address space
- Isolates address spaces
  - One process can't interfere with another's memory
  - User program cannot access privileged system information

VM as a Tool for Caching

- *Virtual memory* is an array of *N* contiguous bytes stored on disk.
- Some contents on disk are cached in *physical memory* (*DRAM cache*)
  - These cache blocks are called *pages* (size is $P = 2^p$ bytes)



DRAM Cache Organization

- Driven by the enormous miss penalty
  - DRAM is about ***10x*** slower than SRAM
  - Disk is about ***10,000x*** slower than DRAM
- Consequences
  - Large page (block) size: typically 4-8 KB
  - Fully associative:
    - Any VP can be placed in any PP
    - Requires a "large" mapping function
  - Highly sophisticated, expensive replacement algorithms
  - *Write-back* rather than *write-through*