CS420: Operating Systems

Introduction

James Moscola
Department of Engineering & Computer Science
York College of Pennsylvania

What is an Operating System?

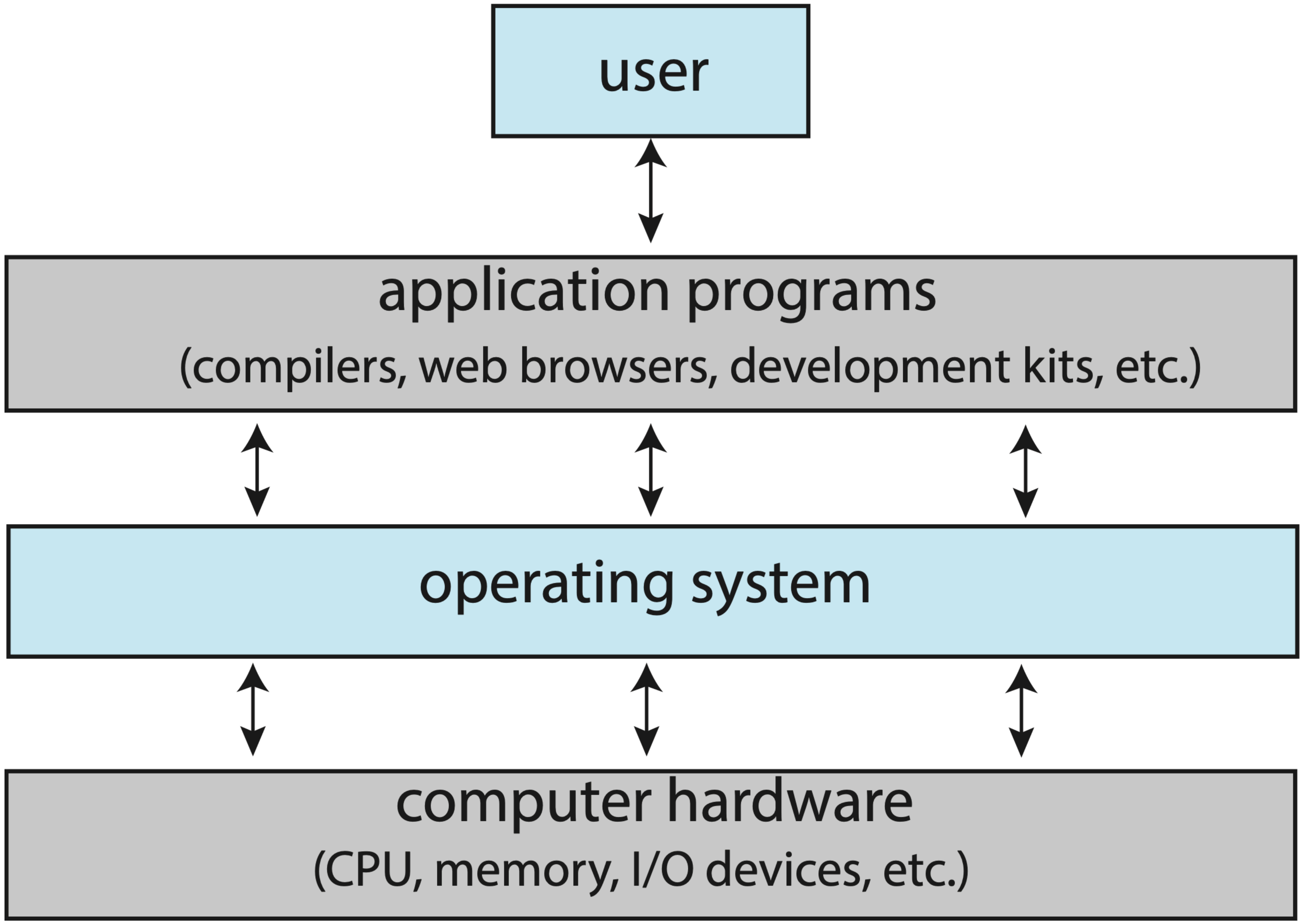
- **A program that acts as an intermediary between a user of a computer and the computer hardware**

- **Operating system goals:**
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Computer System Structure

- **Computer system can be divided into four main components:**
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers

Components of a Computer System



What Do Operating Systems Do?

- **Depends on who is using the OS**
  - Users want convenience, ease of use
    - Don't care about resource utilization
  - However, a shared computer, such as mainframe or minicomputer, must keep all users happy
  - Users of dedicated systems such as workstations have dedicated resources but frequently use shared resources from servers
  - Handheld computers are resource poor, optimized for usability and battery life
  - Some computers have little or no user interface, such as embedded computers in devices and automobiles

Operating System Definition

- **OS is a resource allocator**
  - Manages all system resources
  - Decides between conflicting requests for efficient and fair resource use

- **OS is a control program**
  - Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition (Cont.)

- **No universally accepted definition**

- **"Everything a vendor ships when you order an operating system" is good approximation**
  - But varies wildly

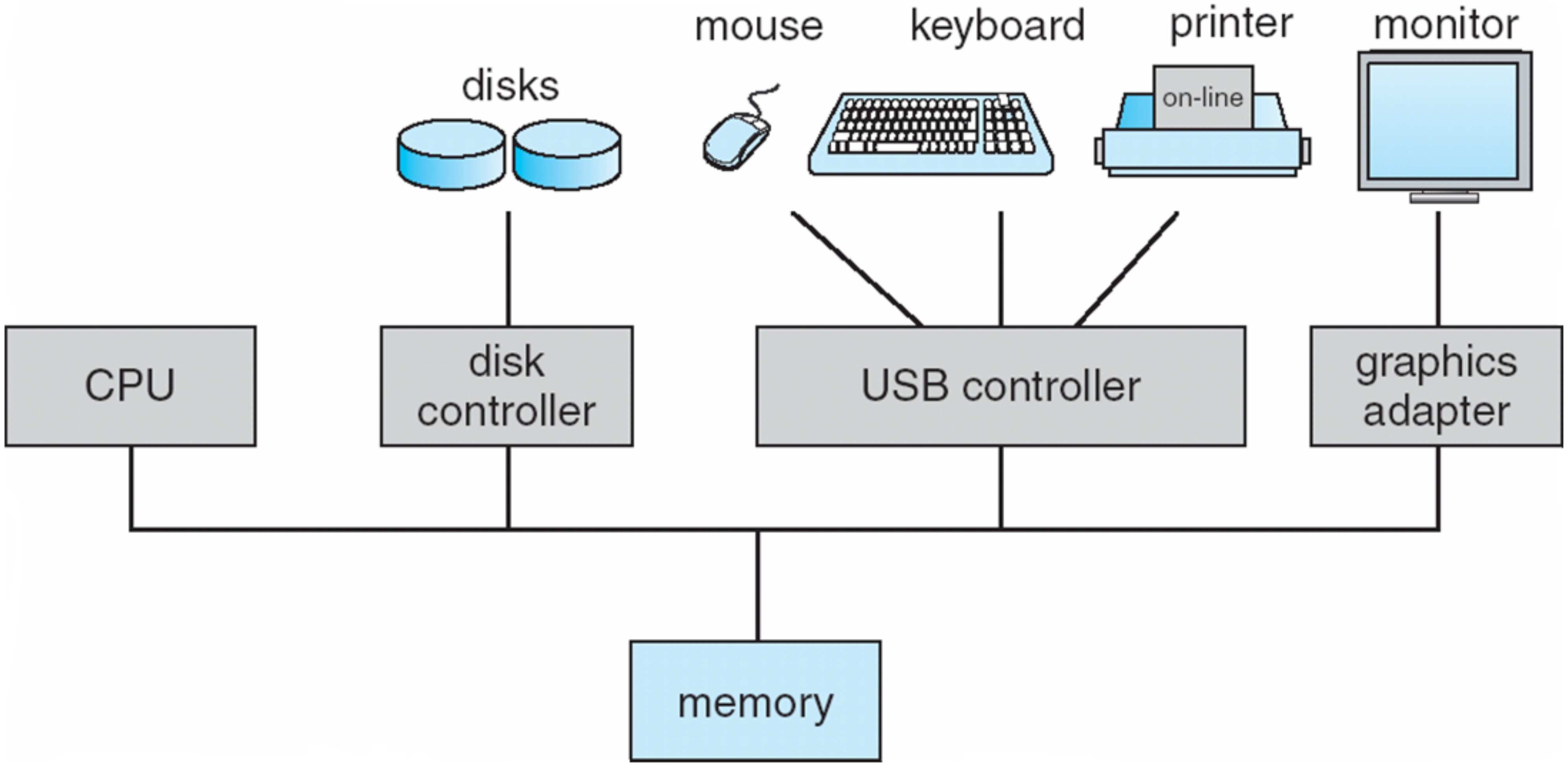
- **"The one program running at all times on the computer" is the kernel. Everything else is either a system program (ships with the operating system) or an application program.**

Computer Startup / System Boot

- **Bootstrap program is loaded at power-up or reboot**
  - Typically stored in ROM or EPROM, generally known as firmware
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution
- **Operating system must be made available to hardware so hardware can start it**
  - A bootstrap loader, locates the kernel, loads it into memory, and starts it
  - Sometimes two-step process where boot block at fixed location loads bootstrap loader
  - When power initialized on system, execution starts at a fixed memory location
    - Firmware used to hold initial boot code

Computer System Organization

- **Computer-system operation**
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



Computer System Operation

- **I/O devices and the CPU can execute concurrently**
- **Each device controller is in charge of a particular device type**
- **Each device controller has a local buffer**
  - CPU moves data from/to main memory to/from local buffers
  - I/O is from the device to local buffer of controller
- **Device controller informs CPU that it has finished its operation by causing an interrupt**

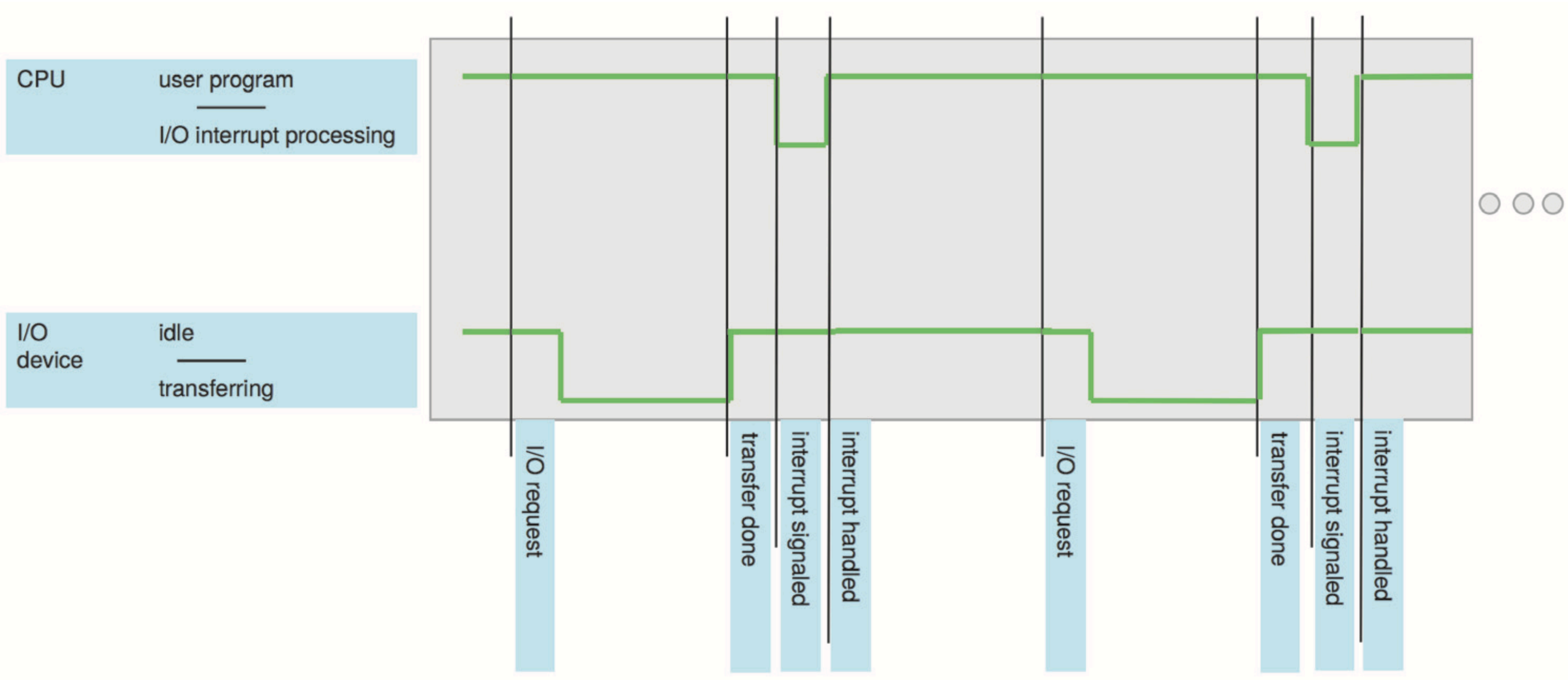
Common Functions of Interrupts

- **Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the addresses of all the service routines**
- **Interrupt architecture must save the address of the interrupted instruction**
- **Incoming interrupts are disabled while another interrupt is being processed to prevent a lost interrupt**
- **A trap is a software-generated interrupt caused either by an error or a user request**
- **An operating system is interrupt driven**

Interrupt Handling

- **The operating system preserves the state of the CPU by storing registers and the program counter**
- **Separate segments of code determine what action should be taken for each type of interrupt**

Interrupt Timeline for a Single Program



I/O Structure

- **After I/O starts, control returns to user program only upon I/O completion**
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing
- **After I/O starts, control returns to user program without waiting for I/O completion**
  - System call – request to the operating system to allow user to wait for I/O completion
  - Device-status table contains entry for each I/O device indicating its type, address, and state
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt

Direct Memory Access Structure

- **Used for high-speed I/O devices able to transmit information at close to memory speeds**
- **Device controller transfers blocks of data from local storage buffer directly to main memory without CPU intervention**
- **Only one interrupt is generated per block, rather than the one interrupt per byte**

Storage Structure

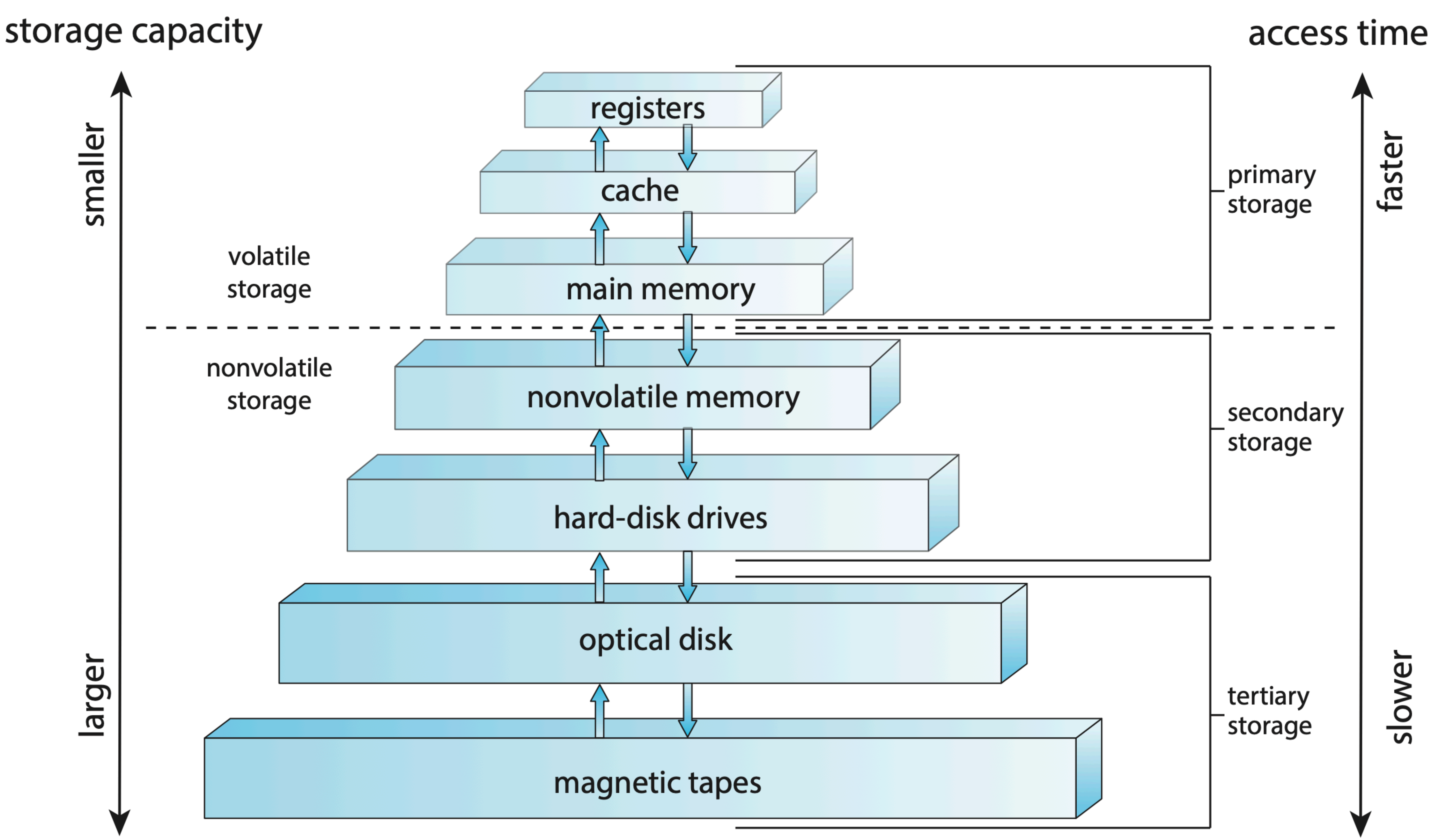
- **Main memory – only large storage media that the CPU can access directly**
  - Random access
  - Typically volatile (i.e. loses contents when powered off)
- **Secondary storage – extension of main memory that provides large nonvolatile storage capacity**
  - Magnetic hard disk drives (HDDs) – rigid metal or glass platters covered with magnetic recording material
    - Disk surface is logically divided into tracks, which are subdivided into sectors
    - The disk controller determines the logical interaction between the device and the computer
  - Solid State Disks (SSDs) are quickly replacing magnetic disks
    - Cost/MB is still higher than magnetic disks

Storage Hierarchy

- **Storage systems organized in hierarchy**
  - Speed
  - Cost
  - Volatility

- **Caching – copying information into faster storage system; main memory can be viewed as a cache for secondary storage**

Storage-Device Hierarchy



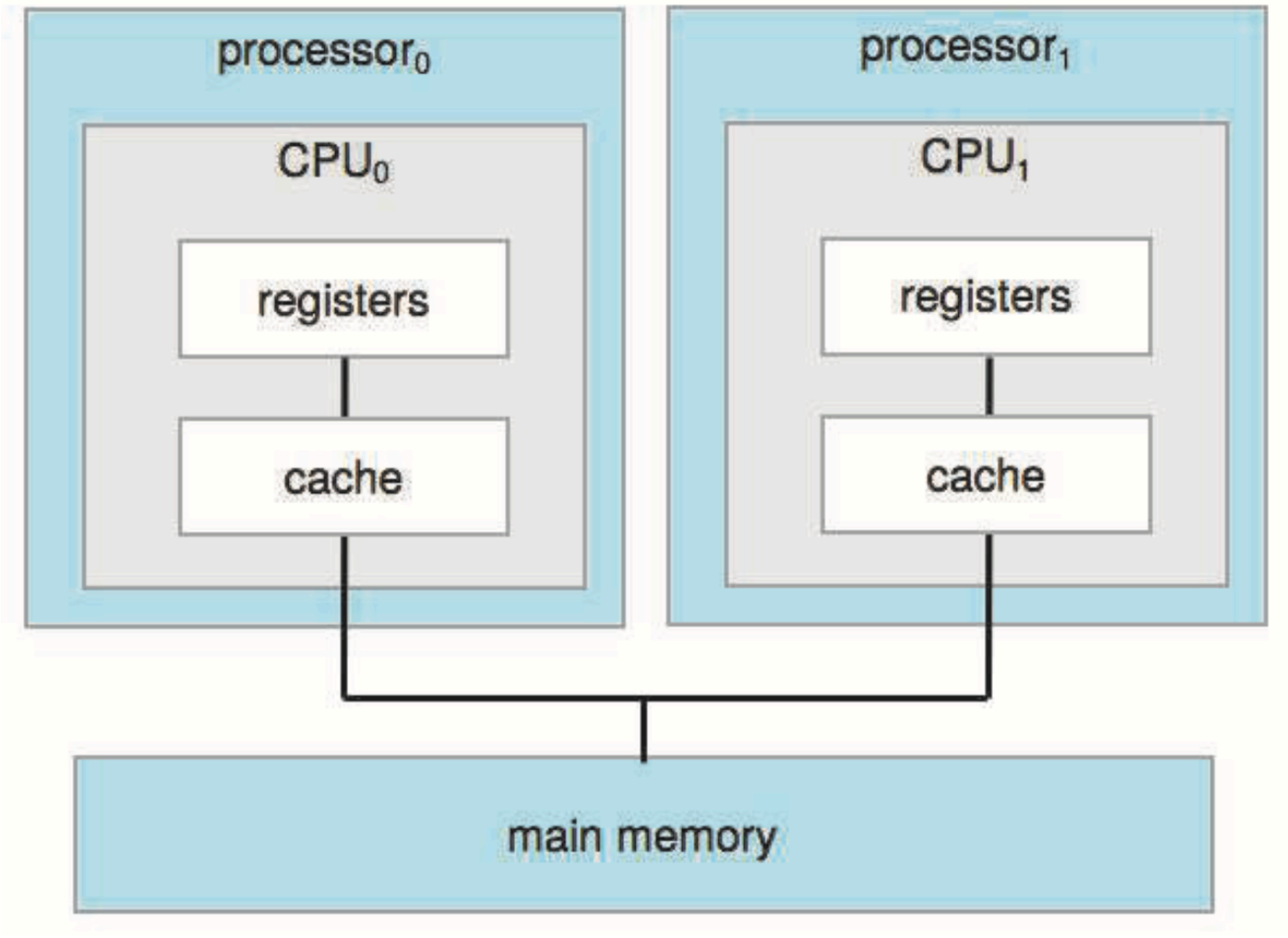
Caching

- **Important principle, performed at many levels in a computer (in hardware, operating system, software)**
- **Information in use copied from slower to faster storage temporarily**
- **Faster storage (cache) checked first to determine if information is there**
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- **Cache smaller than storage being cached**
  - Cache management important design problem
  - Cache size and replacement policy

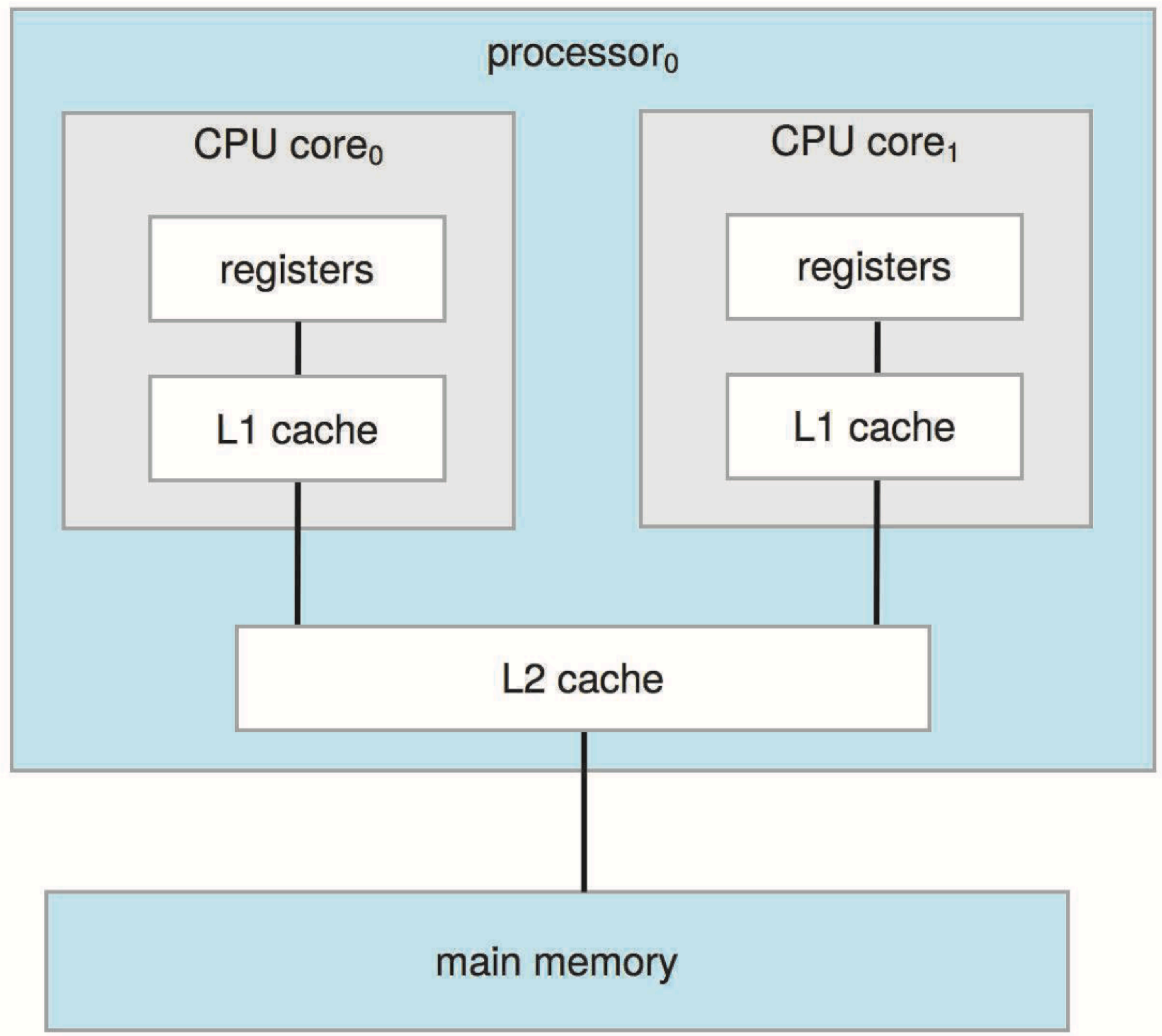
Computer-System Architecture

- **Most modern computing systems utilize a single general-purpose processor (typically with multiple cores)**
  - Most systems also have special-purpose processors as well (e.g. disk controllers, USB controllers, DMA controllers, etc.)
- **Multiprocessors / multicore processors now present in most systems**
  - Once need multiple physical processors to achieve multiple cores, now many cores are available on a single processor die
  - Advantages include:
    - Increased throughput
    - Economy of scale
    - Increased reliability – graceful degradation or fault tolerance
  - Two approaches to using multiprocess/multicore systems:
    - Symmetric Multiprocessing - each processor performs all tasks
    - Asymmetric Multiprocessing - each processor is assigned a specific task

Symmetric Multiprocessing Architecture

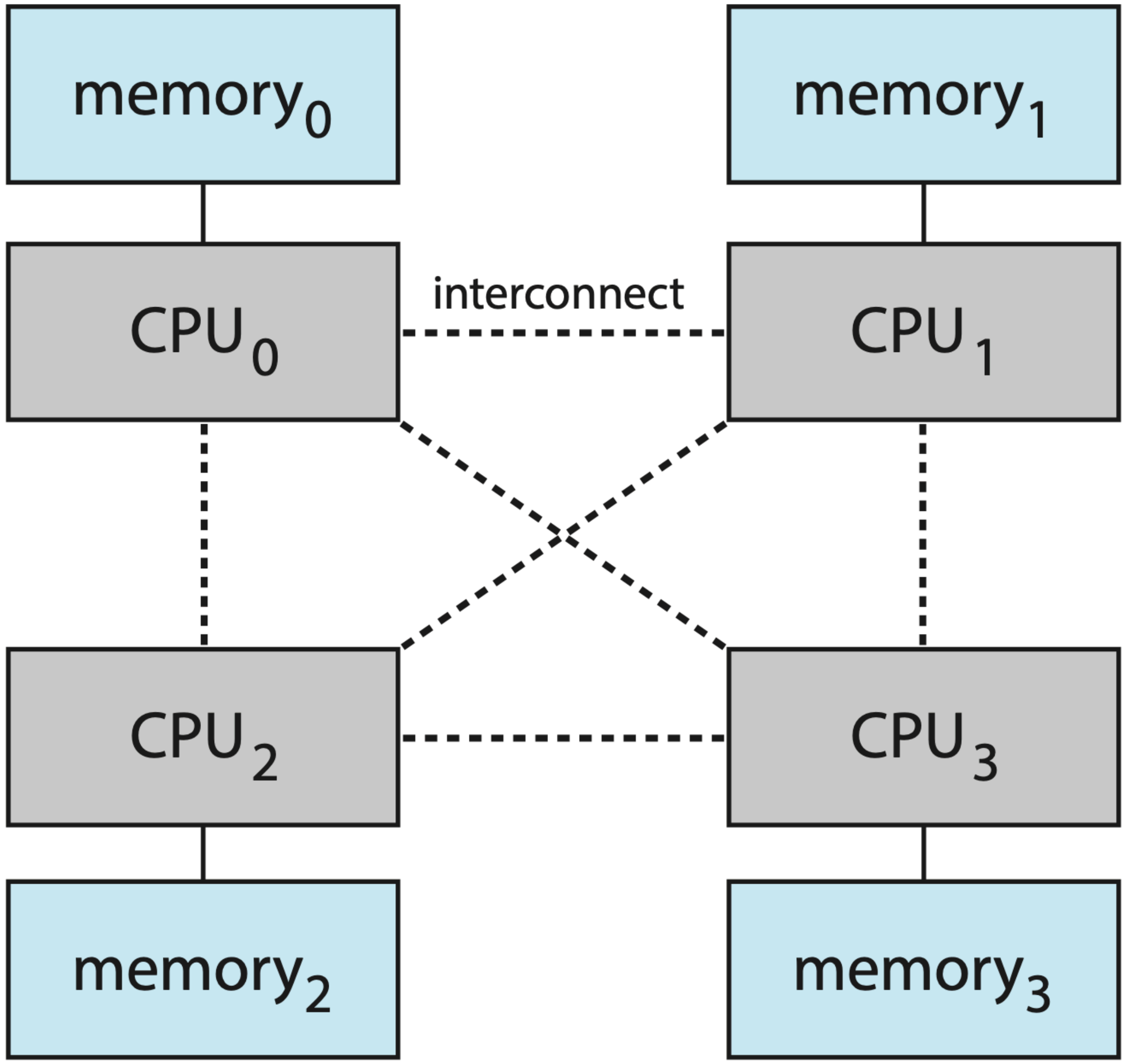


A Dual-Core Design



NUMA Multiprocessing Architecture

- **NUMA - Non-Uniform Memory Access**
  - Reduces bottleneck of ALL CPU cores accessing the same main system memory
  - Each CPU core is given it's own local memory
    - Very fast and minimal contention
  - Each CPU core can access the remote memory connected to other CPU cores via the interconnect
    - Longer latency equates to a penalty when reading non-local memory



Clustered Systems

- **Like multiprocessor systems, but multiple systems working together**
  - Usually sharing storage via a storage-area network (SAN)
  - Provides a high-availability service which survives failures
    - Asymmetric clustering has one machine in hot-standby mode
    - Symmetric clustering has multiple nodes running applications, monitoring each other
  - Some clusters are for high-performance computing (HPC)
    - Applications must be written to use parallelization

General Structure of a Clustered System