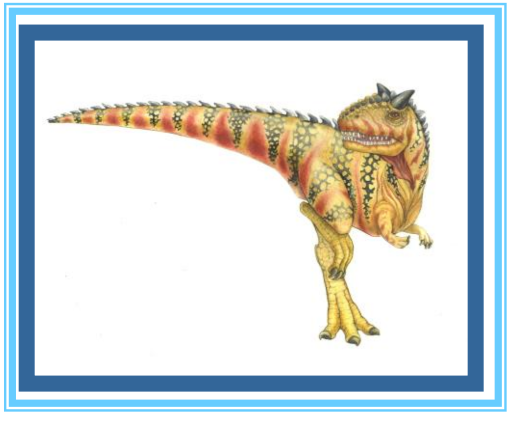
Chapter 1: Introduction
Part I



Chapter 1: Introduction

- What Operating Systems Do
- Computer-System Organization
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Distributed Systems
- Special-Purpose Systems
- Computing Environments
- Open-Source Operating Systems

Objectives

- To provide a grand tour of the major operating systems components
- To provide coverage of basic computer system organization

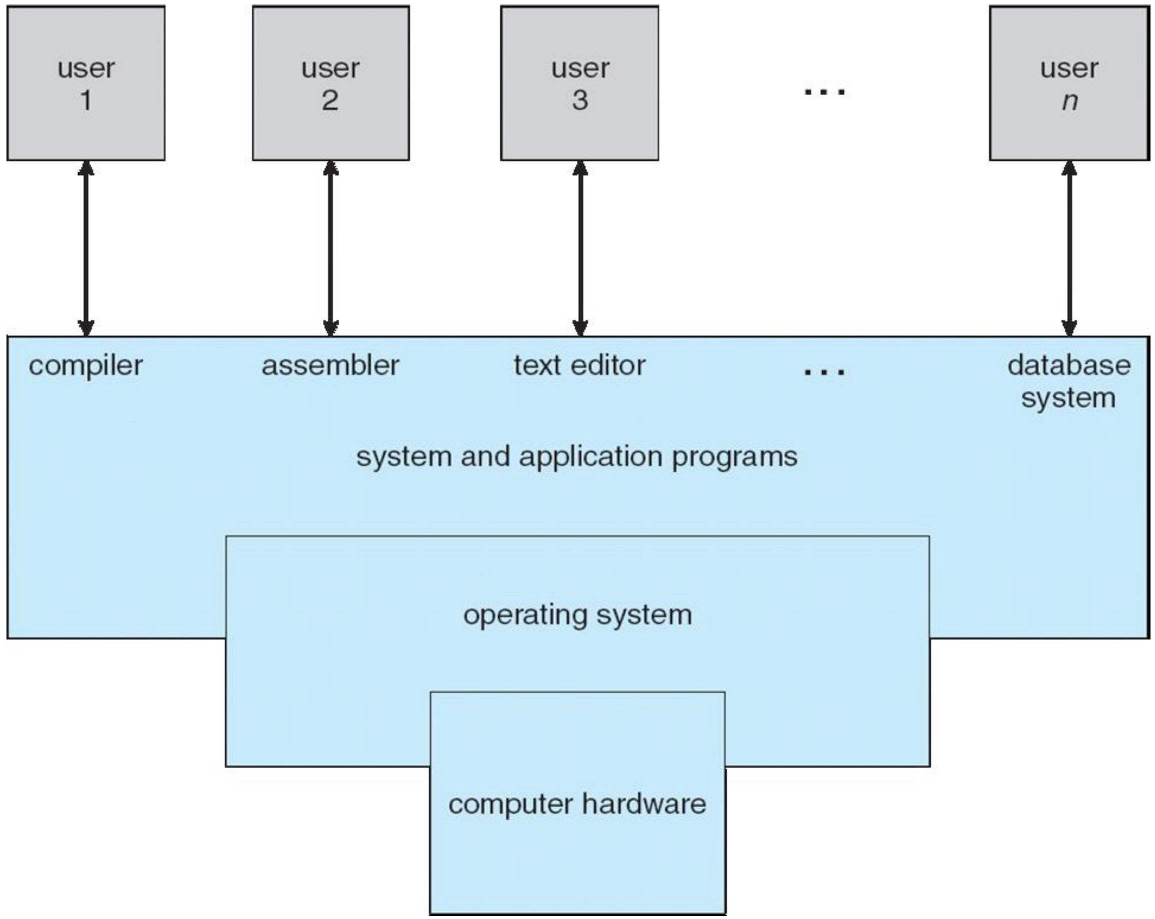
What is an Operating System?

- 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 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

Four Components of a Computer System



Operating System Definition

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

1. Give a specific example of resource management
2. Give an example of conflicting requests

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

1. Give a couple of examples of how an OS prevents errors and improper use

Operating System Definition (Cont)

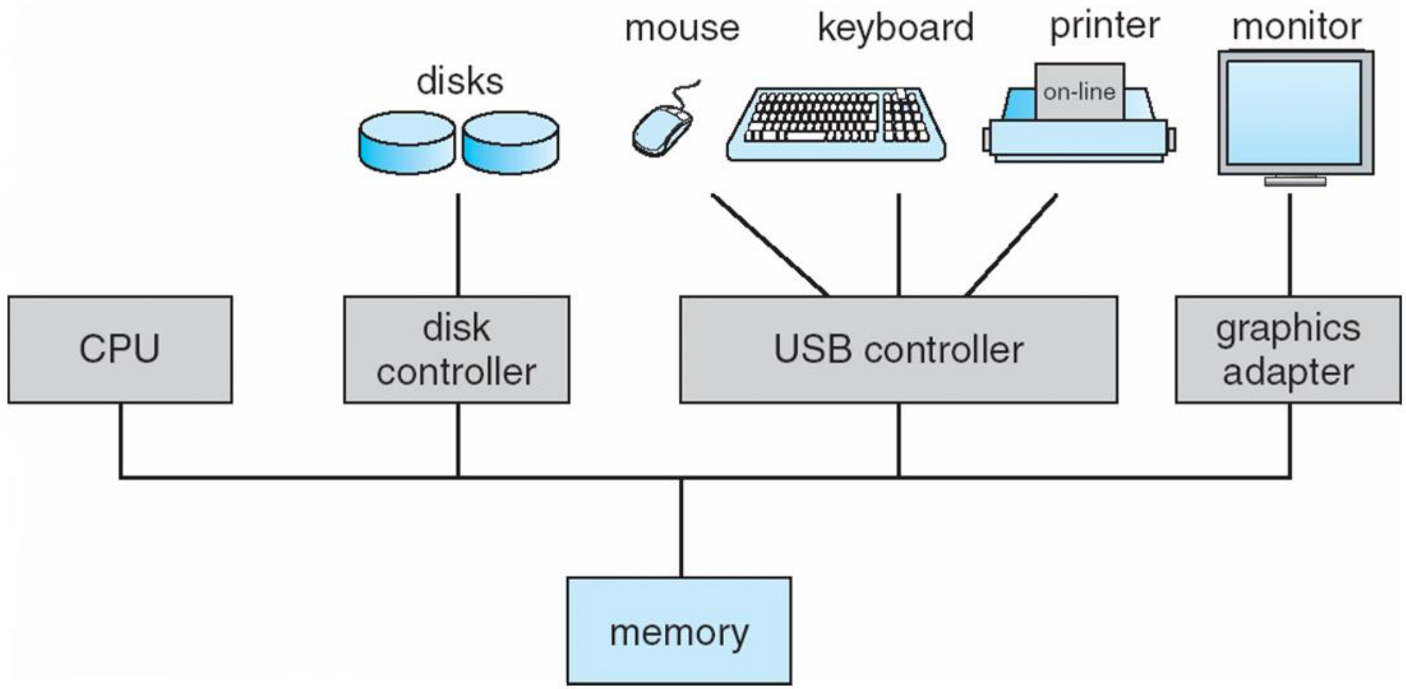
- No universally accepted definition
- "The one program running at all times on the computer" is the **kernel**. Everything else is either
  a. a system program (ships with the operating system) or
  b. an application program

Computer Startup

- **bootstrap program** is loaded at power-up or reboot
  1. Where is the bootstrap program found?
  2. What does the bootstrap program do?
  3. Can the bootstrap program of the computer you own be modified?

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*

1. Let’s trace through a simple request for data from the hard drive from a running program. What happens?

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
  1. Let's look at a picture
- Interrupt architecture must save the address of the interrupted instruction
  1. Why? What else must be saved?
- Incoming interrupts are *disabled*
  1. Why?
- A *trap* is a software-generated interrupt caused either by an error or a user request
  1. Any ideas what kind of errors cause a trap?
- An operating system is **interrupt driven**
  1. Why is an interrupt driven OS essential?

Interrupt Handling

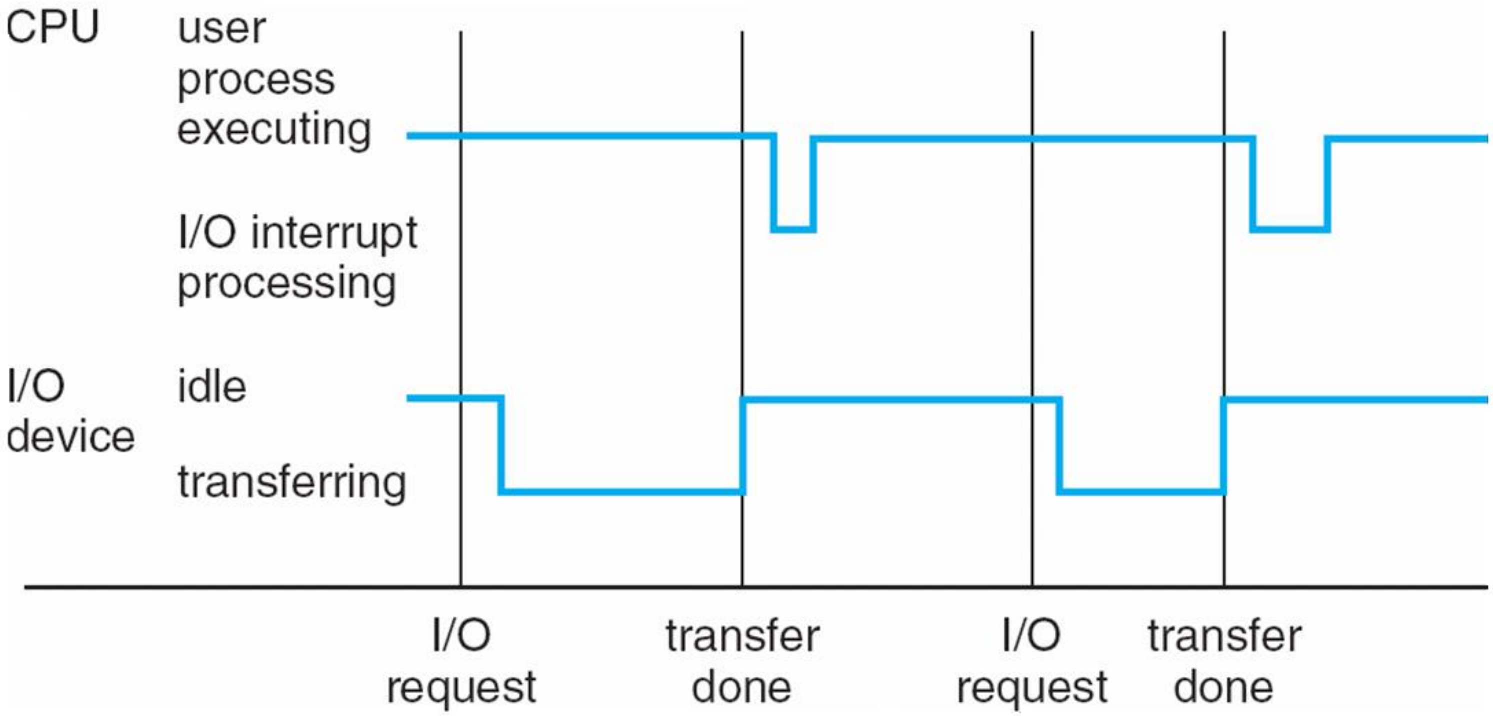
- The operating system preserves the state of the CPU by storing registers and the program counter

1. Where?

- Separate segments of code determine what action should be taken for each type of interrupt

1. How are the segments of code found?

Interrupt Timeline



Interrupt timeline for a single process doing output.

1. Explain the diagram

I/O Structure

- General-purpose computers consist of CPUs & multiple device controllers connected through a common bus
- Device controller is bridge between OS and device
- Device driver is part of OS (some will debate not) that provides communication with device controller
- I/O operation
    1. device driver loads device controller registers
    2. controller examines registers to determine action
    3. controller starts data transfer to/from local buffer
    4. sends interrupt upon completion
- OK for moving small amounts of data

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 buffer storage directly to main memory without CPU intervention
- Only one interrupt is generated per block, rather than the one interrupt per byte for low-speed devices

Storage Structure

- Main memory – only large storage media that the CPU can access directly
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity
- Magnetic disks – 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
- "Seagate reportedly began shipping the industry's first 4 TB-class hard drives with 1 TB per platter density. Slotted in the company's Barracuda 7200.15 series, the drive provides 4000 GB of unformatted space, backed by 7,200 RPM spindle-speed, 64 MB buffer, and SATA 6 Gb/s interface"

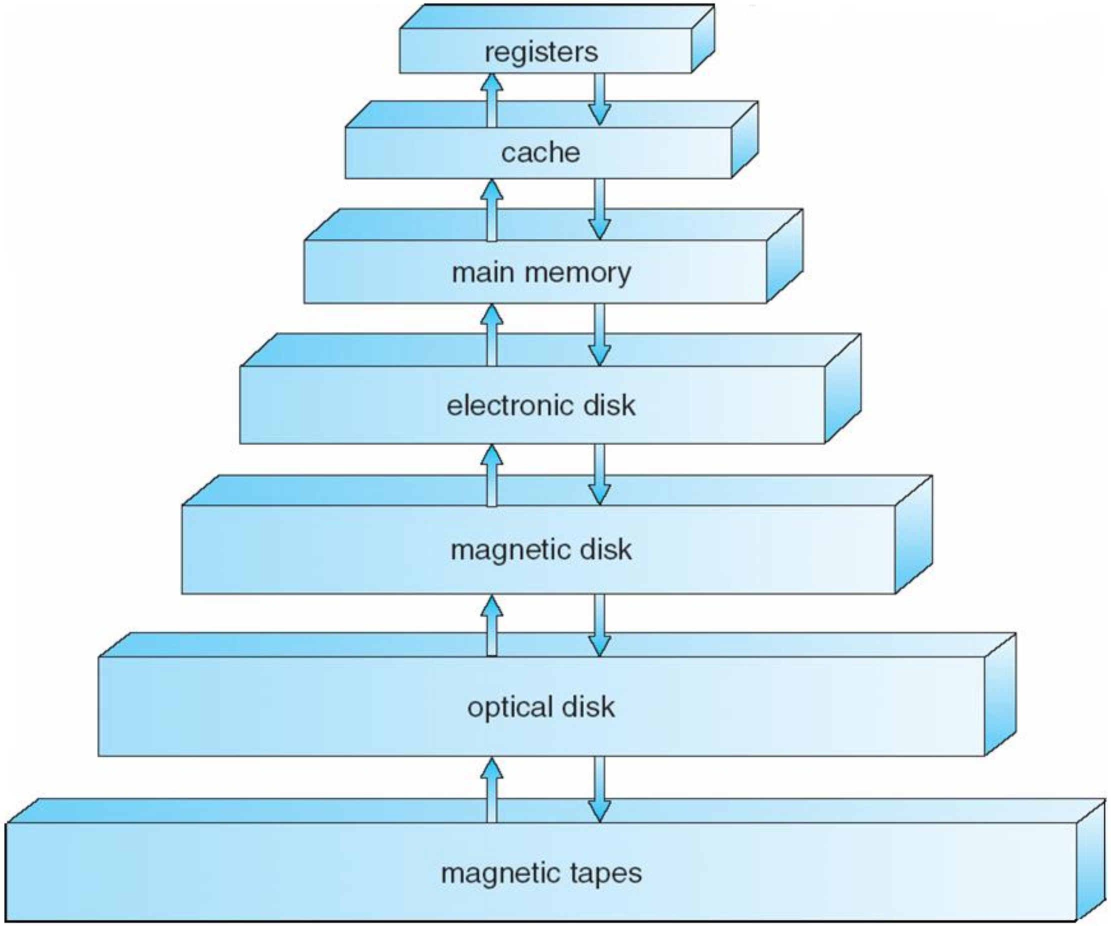
http://www.techpowerup.com/182318/seagate-ships-4-tb-class-hard-drives-with-1-tb-per-platter-density.html

Storage Hierarchy

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

- **Caching** – copying information into faster storage system; main memory can be viewed as a last *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 systems use a single general-purpose processor (mobile devices through mainframes)
  - Most systems have special-purpose processors as well (e.g. graphics processors)

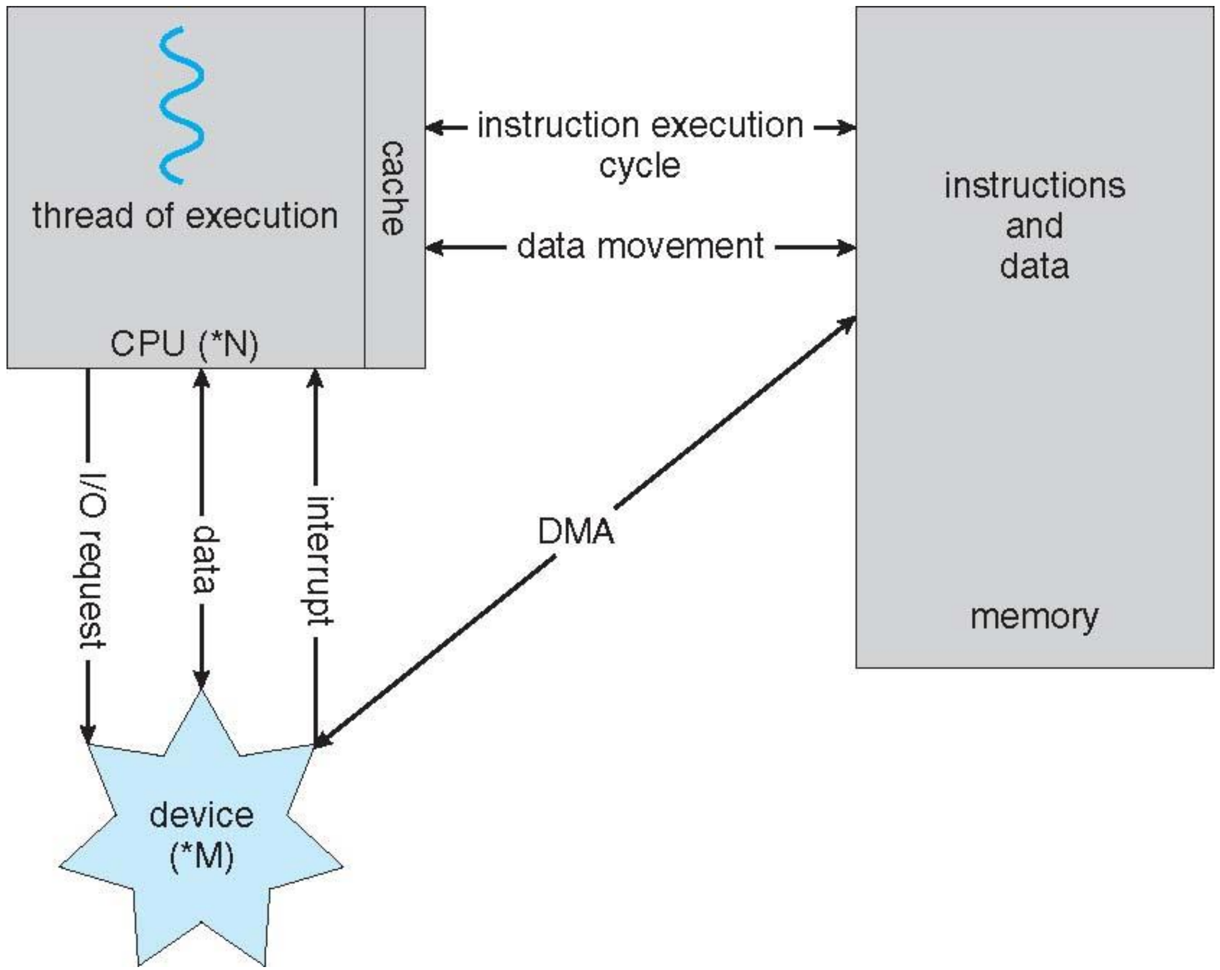
- CPU - few cores optimized for sequential processing
- GPU - hundreds to thousands of smaller cores for handling simultaneous tasks

1. What is happening with graphics processors in today's systems?

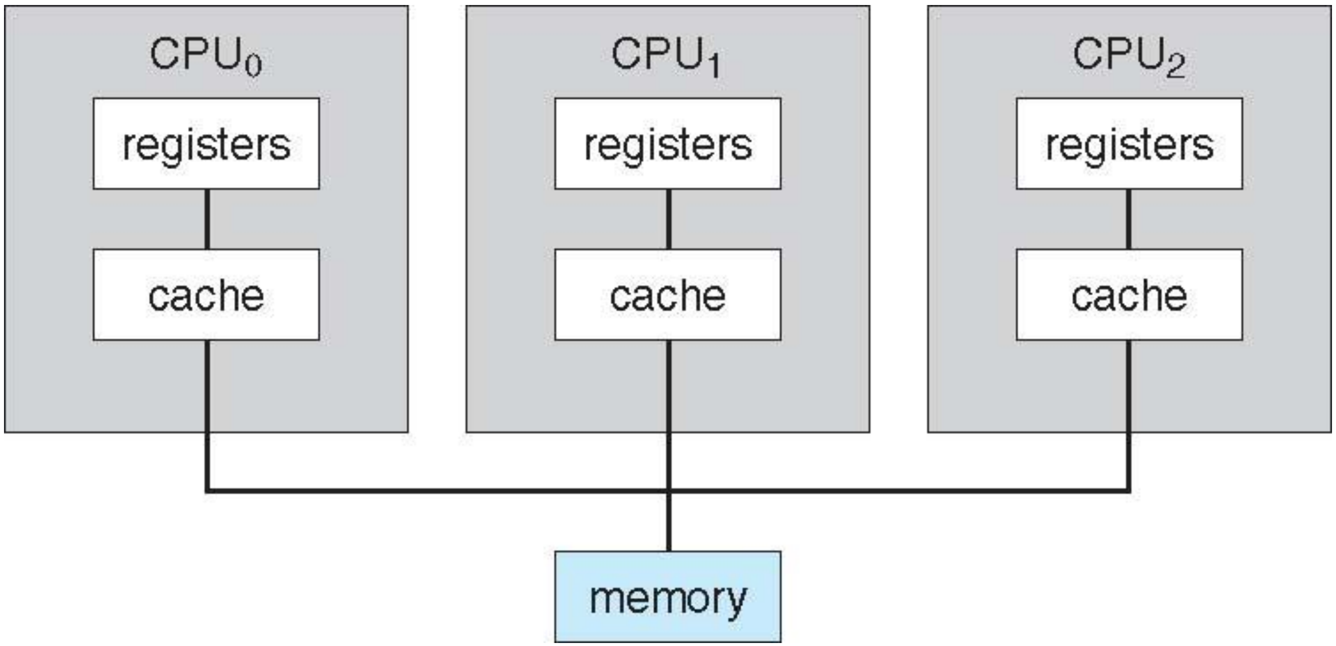
Computer-System Architecture

- Multiprocessors systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems, multicore systems)
  - Advantages include
    1. Increased throughput
    2. Economy of scale - single multiprocessor system typically cost less than multiple single processor systems and can share data more efficiently
    3. Increased reliability – graceful degradation or fault tolerance … if one processor fails others can pick up the load without shutting entire system down
  - Two types
    1. Asymmetric Multiprocessing - Boss-Worker
    2. Symmetric Multiprocessing (SMP) - all processors are peers

How a Modern Computer Works

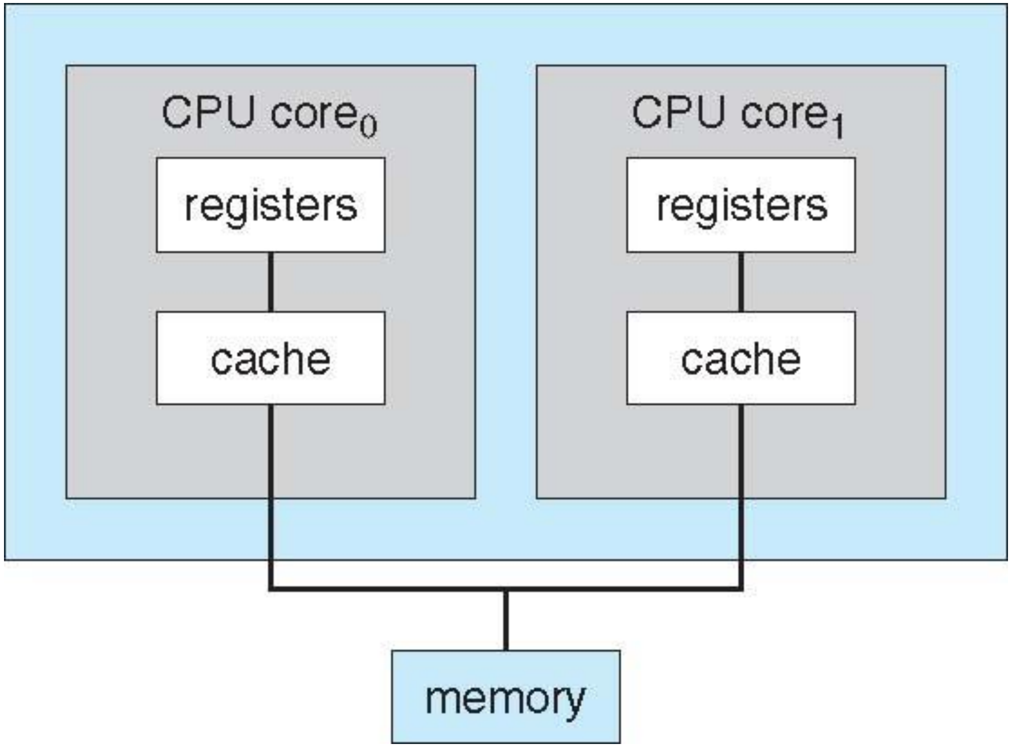


Symmetric Multiprocessing Architecture



1. Give one major pro and one major con to this architecture

A Dual-Core Design



1. T / F All multicore systems are multiprocessor systems
2. T / F All multiprocessor systems are multicore