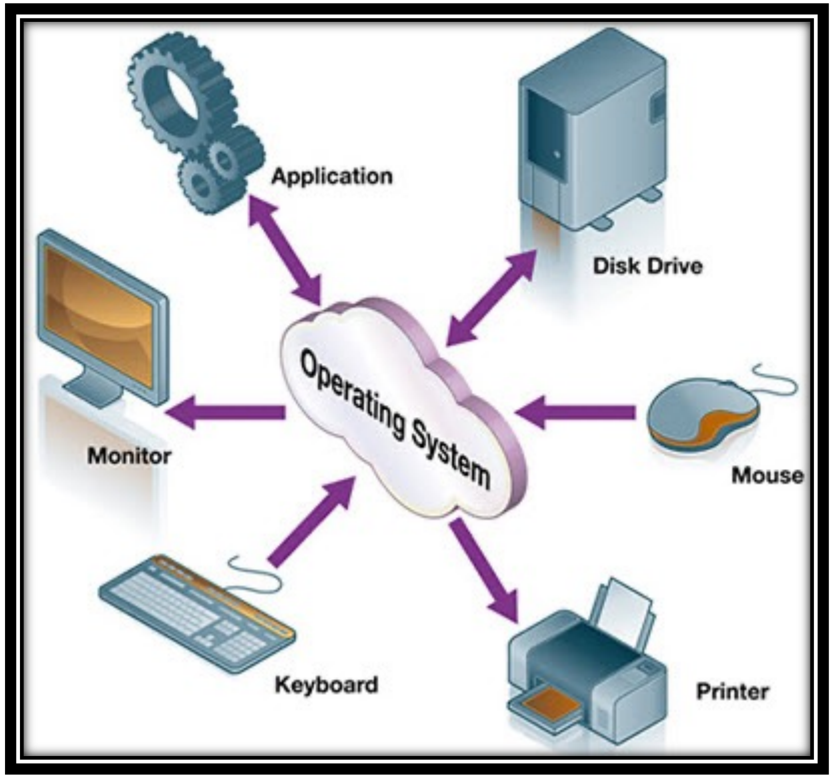
BITS Pilani
Hyderabad Campus

OPERATING SYSTEMS (CS F372)

Introduction

Dr. Barsha Mitra
CSIS Dept., BITS Pilani, Hyderabad Campus

What is an Operating System



Handout Overview

Objectives

- To learn about how process management is carried by the OS. This will include process creation, thread creation, CPU scheduling, process synchronization and deadlocks.
- To learn about memory management carried out by OS. This will include the concepts of paging, segmentation, swapping, and virtual memory.
- To learn how permanent storage like files and disks are managed by OS. This will include topics related to access methods, mounting, disk scheduling, and disk management.
- Hands-on experience

Handout Overview

<u>Text Book:</u>

T1. Silberschatz, Galvin, and Gagne, “Operating System Concepts”, 9th edition, John Wiley & Sons, 2012.

<u>Reference Books:</u>

R1. W. Stallings, “Operating Systems: Internals and Design Principles”, 6th edition, Pearson, 2009.

R2. Tanenbaum, Woodhull, “Operating Systems Design & Implementation”, 3rd edition, Pearson, 2006.

R3. Dhamdhere, “Operating Systems: A Concept based Approach”, 2nd edition, McGrawHill, 2009.

R4. Robert Love, “Linux Kernel Development”, 3rd edition, Pearson, 2010.

Topics to be covered

- Introduction
- OS Structures
- Processes
- Threads
- CPU Scheduling
- Process Synchronization
- Deadlocks
- Main Memory Management
- Virtual Memory
- Mass Storage
- File System Interface
- File System Implementation
- I/O Systems
- Protection

Evaluation

Component	Duration	Weightage (%)	Date & Time	Nature of Component
Mid Semester Examination	90 minutes	30%	As per Time Table	Open Book
Quiz 1	-	10%	TBA	Open Book
Quiz 2	-	10%	TBA	Open Book
Assignment		15%	TBA	Open Book
Comprehensive Examination	120 minutes	35%	As per Time Table	Open Book

Handout Overview

- Chamber Consultation
- Notices
- Make-up Policy

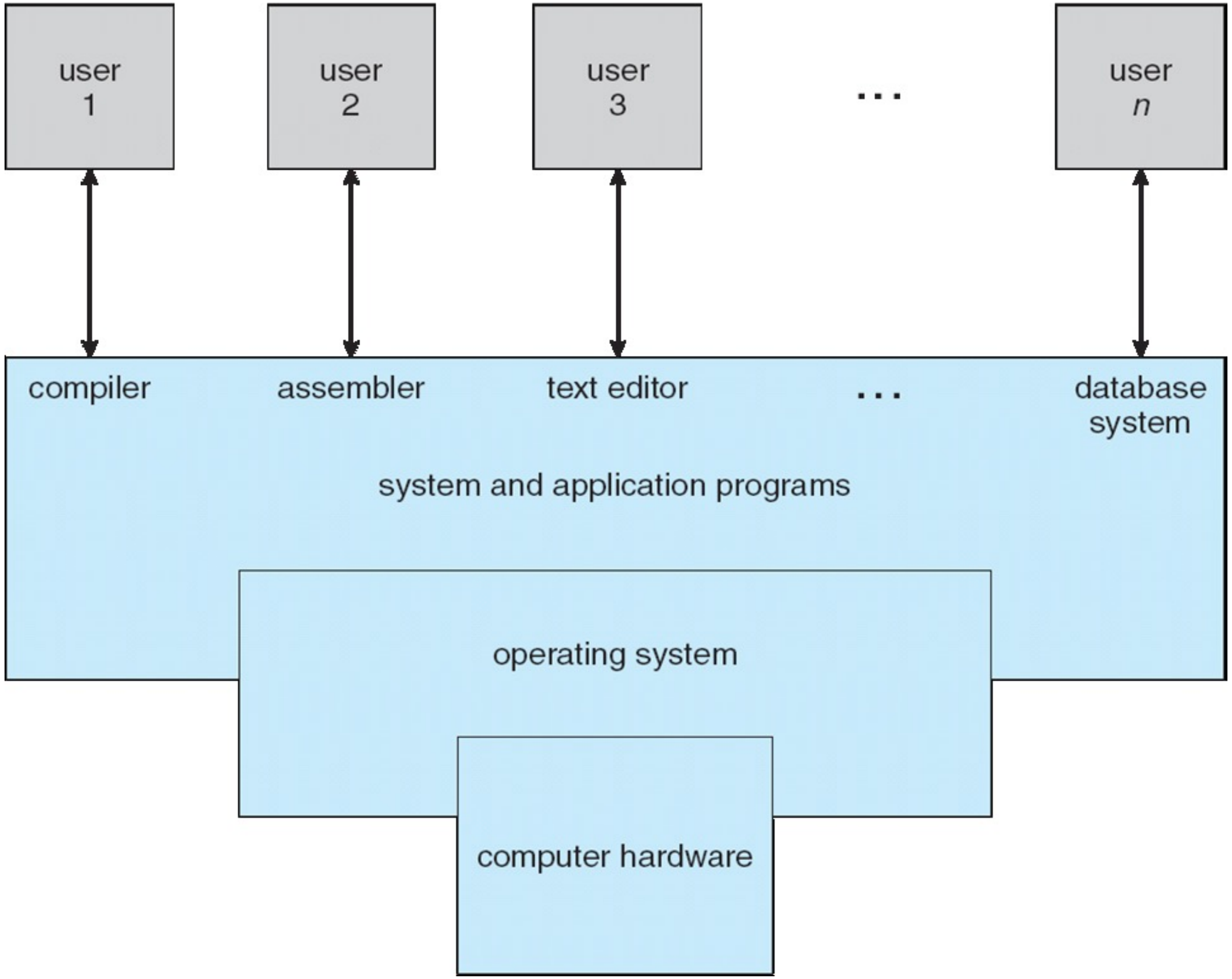
Introduction

- program that manages computer’s hardware
- acts as an intermediary between computer user and computer h/w
- mainframe operating systems
- personal computer (PC) operating systems
- operating systems for mobiles

Computer System Architecture

- Hardware – provides basic computing resources
  - CPU, memory, storage, 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, email, web browsers, database systems, video games, media player
- Users
  - People, machines, other computers

Computer System Architecture



What OS Does? : User View

- Users want **convenience**, **ease of use** and **good performance**
  - Don't care about **resource utilization**
- 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 individual usability and battery life
- Some computers have little or no user interface, such as **embedded computers** in devices and automobiles

What OS Does? : System View

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of user programs to prevent errors and improper use of the computer

How do we define OS?

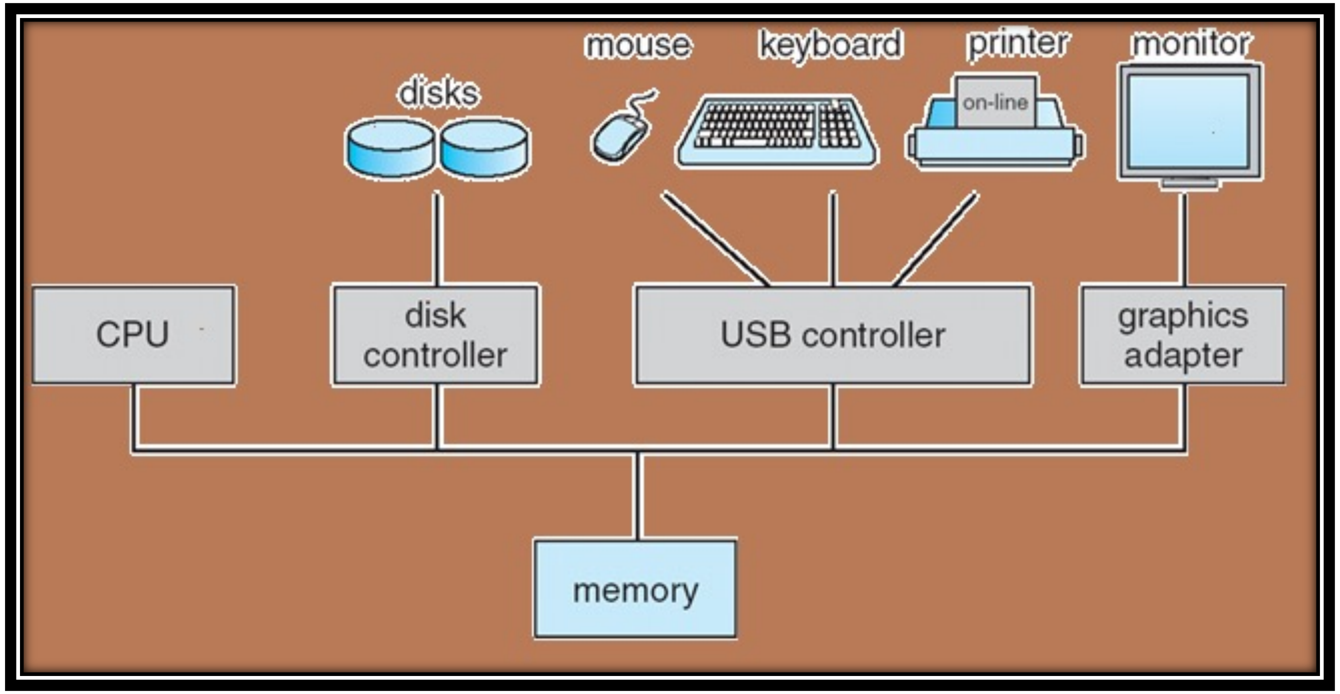
- Everything a vendor ships when you order an operating system is a good approximation

- “The one program running at all times on the computer”

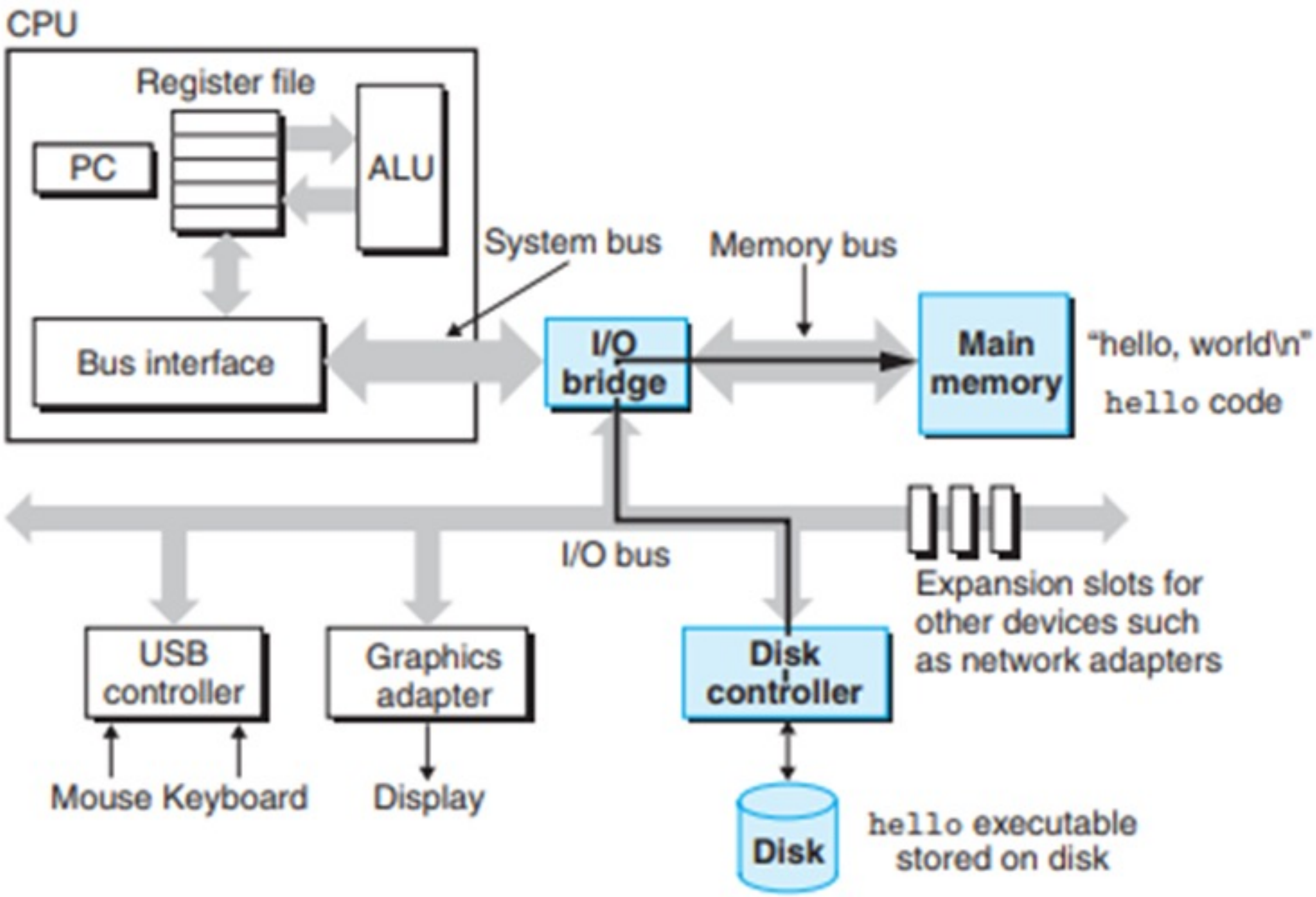
Computer System Organization

- 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

device controller is a hardware component that works as a bridge between the hardware device and the operating system or an application program



Computer-System Organization



Computer-System Operation

- **Bootstrap program** is loaded at power-up or reboot
  - initial program
  - stored in ROM or EPROM, generally known as **firmware**
  - initializes all aspects of system like CPU registers, device controllers, memory contents
  - locates and loads operating system kernel and starts execution

Computer-System Operation

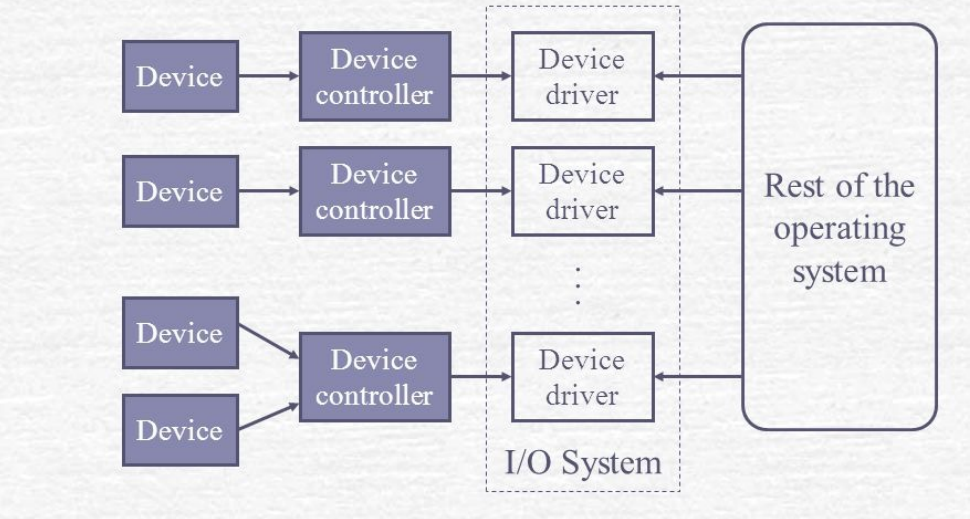
- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs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

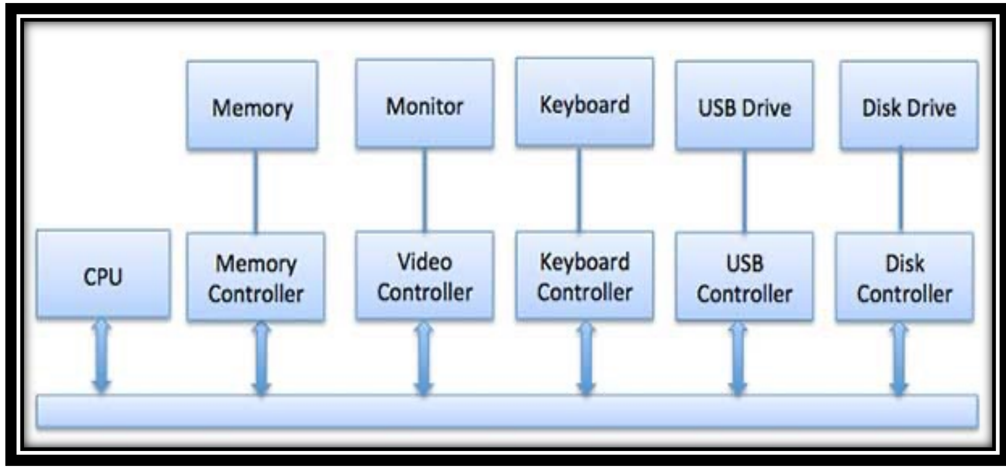
Interrupt Handling

- interrupt transfers control to the interrupt service routine (stored in a fixed location) through the interrupt vector (address for finding ISR)
  - IVT – table of pointers containing the addresses of all the service routines
- must save the address of the interrupted instruction, system stack
- trap/exception is a software-generated interrupt caused either by an error or a user request
- operating system is interrupt driven
- operating system preserves the state of the CPU by storing contents of registers and the program counter

Computer-System Operation

- computer system consists of CPUs and multiple device controllers that are connected through a common bus
- each device controller is in charge of a specific type of device
- device controller
  - maintains some local buffer storage and a set of special-purpose registers
  - moves data between the peripheral devices that it controls and its local buffer storage
- operating systems have a **device driver** for each device controller
- device driver understands the device controller and provides the rest of the operating system with a uniform interface to the device





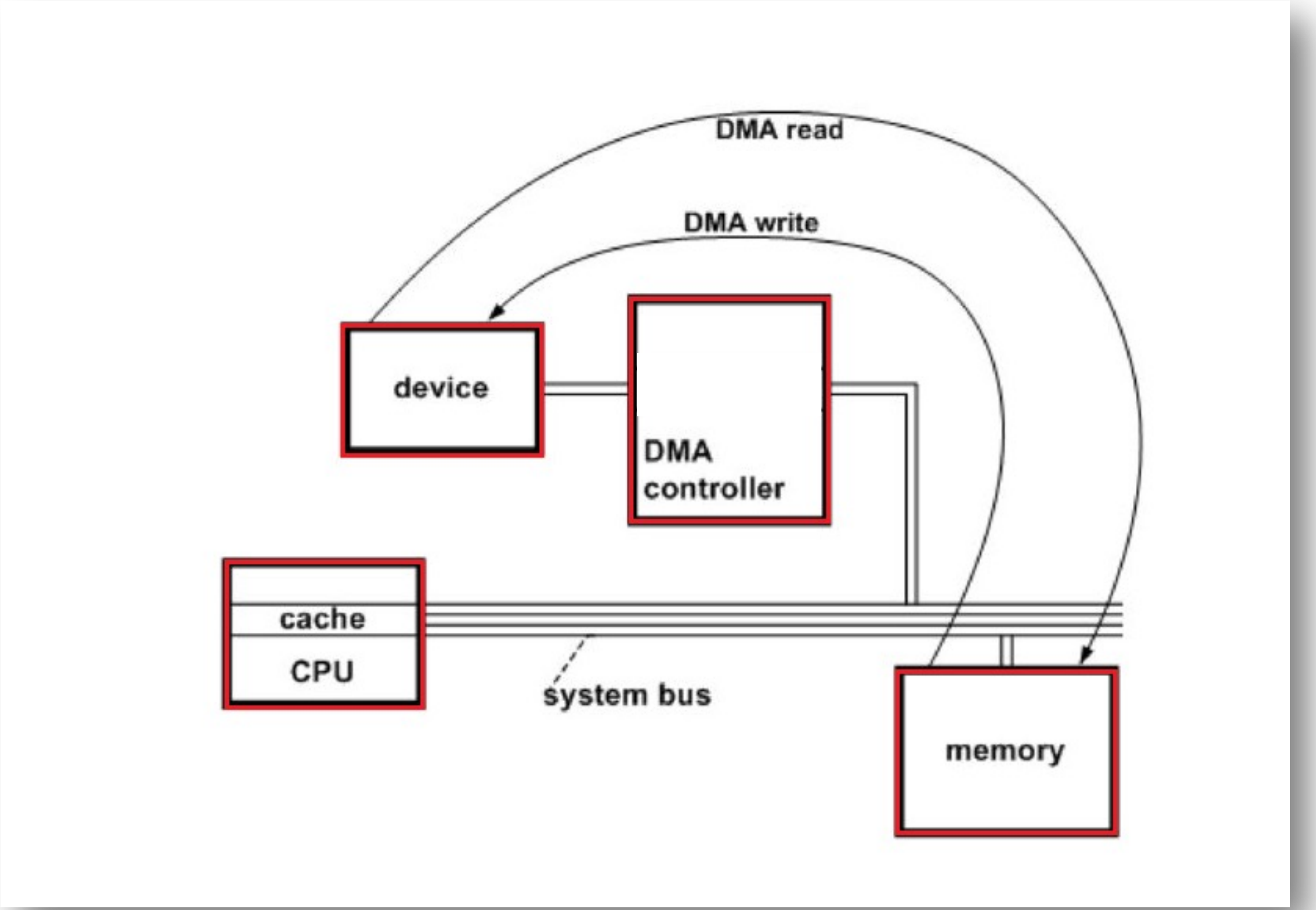
I/O Structure

- to start an I/O operation, the **device driver loads** the registers within the device controller
- **device controller** examines the contents of registers to determine what action to take
- controller starts the transfer of data from the device to its **local buffer**
- device controller informs the **device driver** via an interrupt that it has finished its operation
- device driver then returns control to the operating system, possibly returning the data or a pointer to the data if the operation was a read
- for other operations, the device driver returns status information

I/O Structure

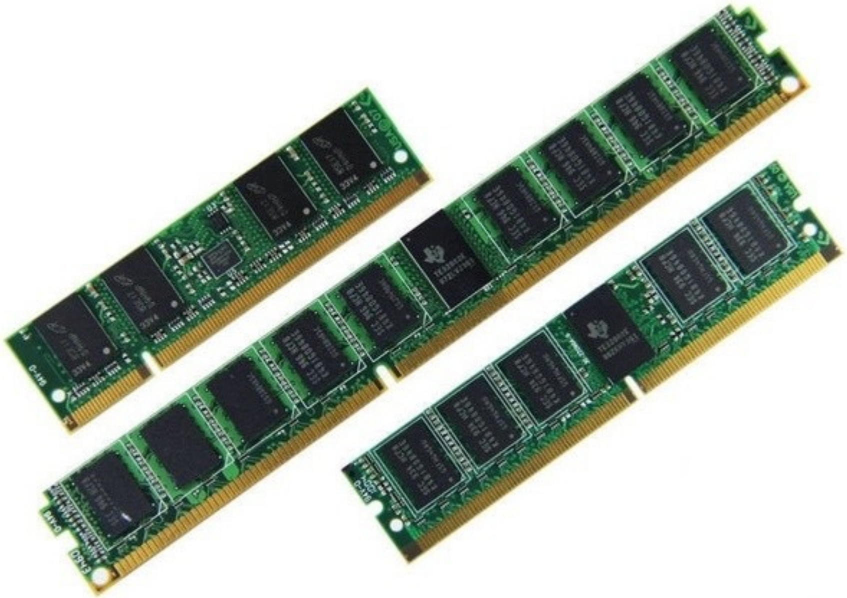
- interrupt-driven I/O is fine for moving small amounts of data
- can produce high overhead when used for bulk data movement such as disk I/O
- **direct memory access (DMA)**
- device controller sets up buffers, pointers, and counters for the I/O device
- device controller transfers an entire block of data directly to or from its own buffer storage to memory, with no intervention by the CPU
- only one interrupt is generated per block, to tell the device driver that the operation has completed
- instead of one interrupt per byte generated for low-speed devices
- CPU is available to accomplish other work

I/O Structure



Storage Structure

- Main memory –
  - only storage media that the CPU can access directly
  - instruction execution
  - random access
  - volatile

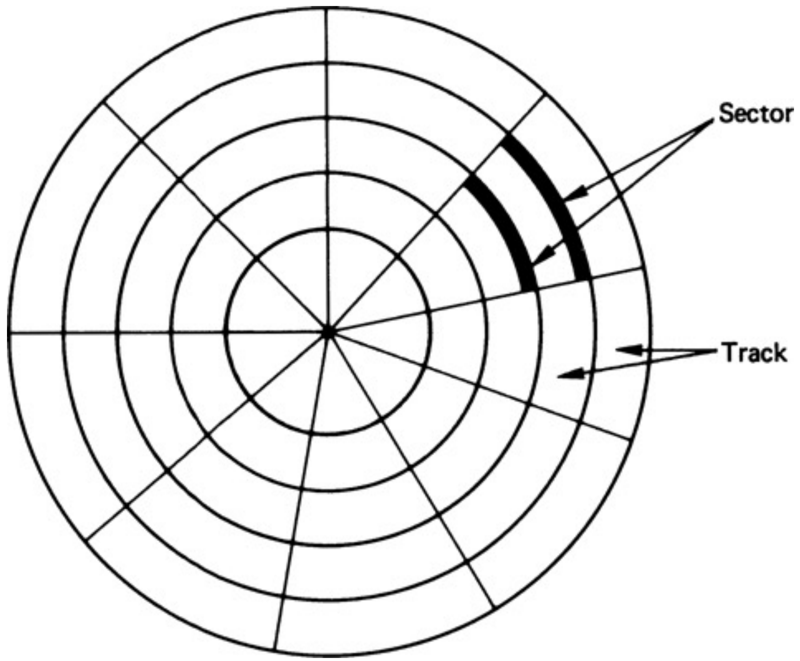


Storage Structure

- Secondary storage –
  - extension of main memory that provides large nonvolatile storage capacity
  - stores both program and data

Storage Structure

- Hard disks/Magnetic disks –
  - rigid metal or glass platters covered with magnetic recording material
  - disk surface is logically divided into tracks, which are subdivided into sectors
  - disk controller determines the interaction between the device and the computer

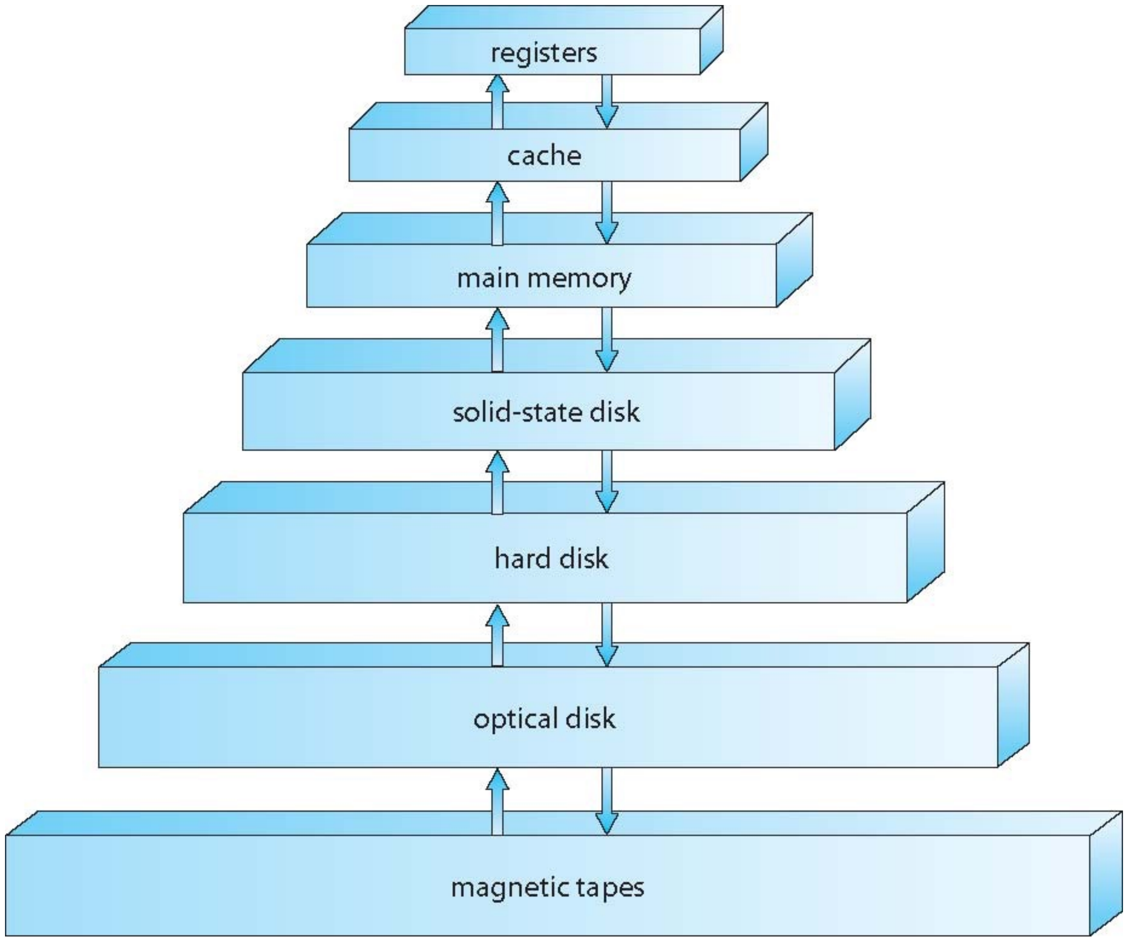


Storage Structure

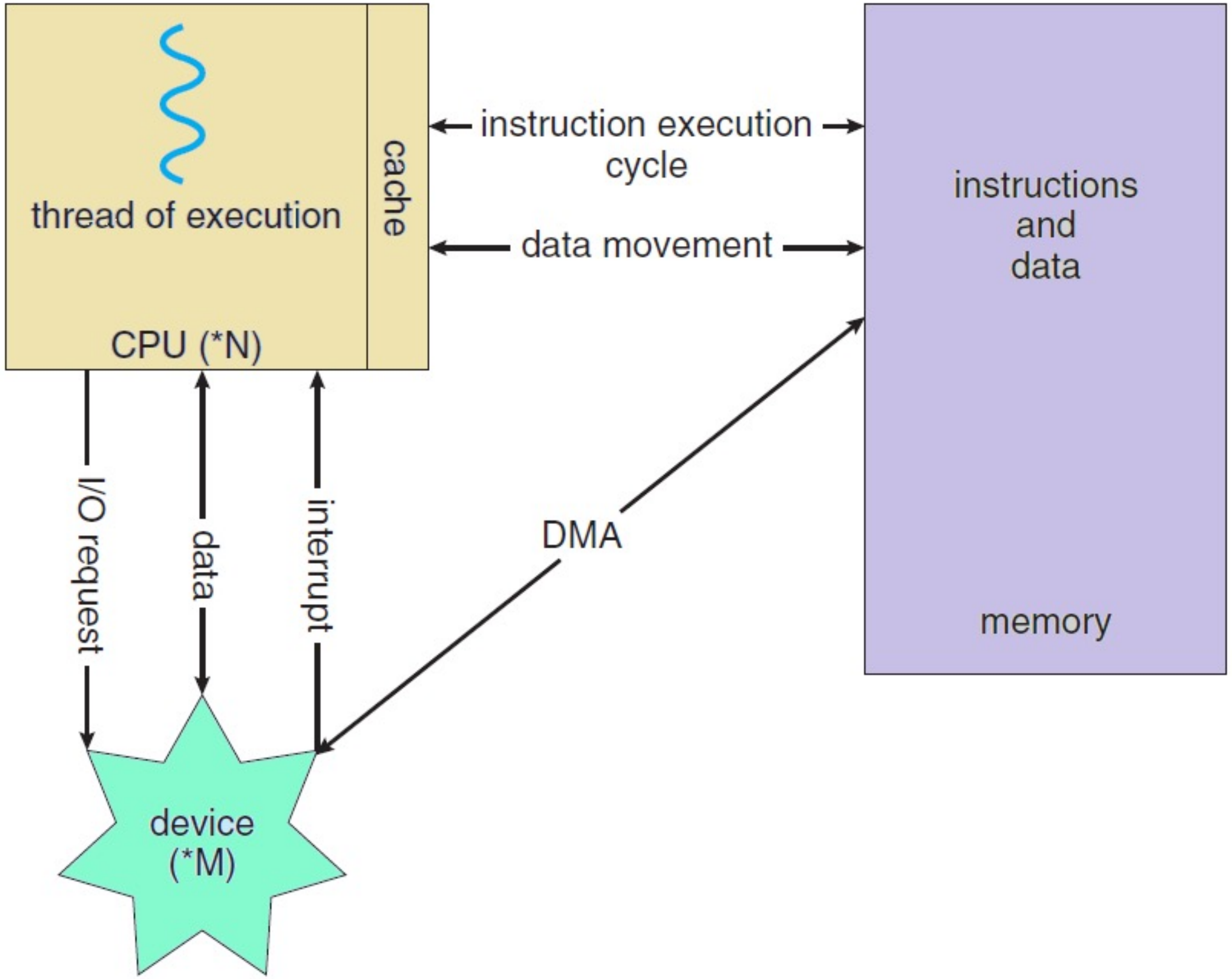
- Solid-state disks –
  - faster than magnetic disks, nonvolatile
  - becoming more popular

- stores data in DRAM during normal operation
- also contains a hidden magnetic hard disk and a battery for backup power
- if external power is interrupted, solid-state disk's controller copies the data from RAM to the magnetic disk
- when external power is restored, the controller copies the data back into RAM
- another form of solid-state disk is flash memory, which is popular in cameras and personal digital assistants (PDAs), slower than DRAM but needs no power to retain its contents

Storage Structure

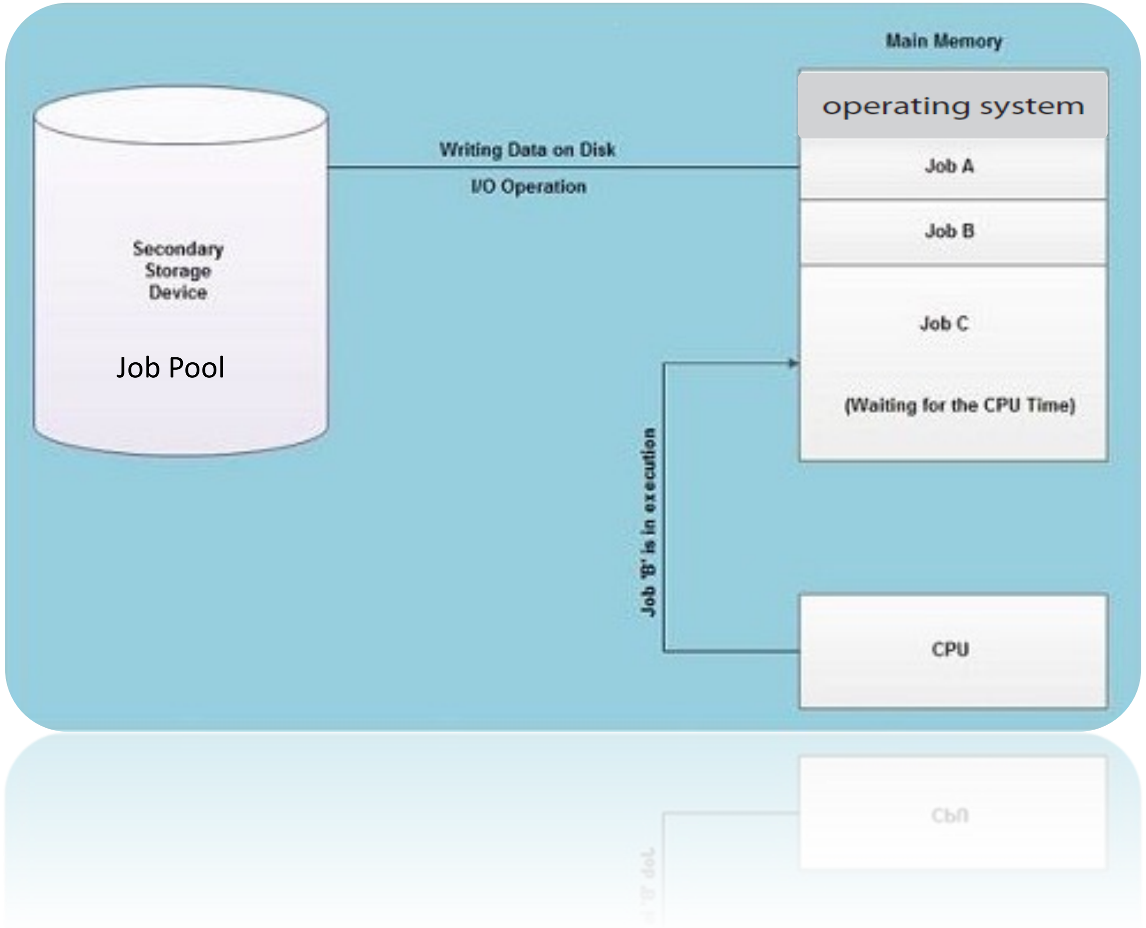


Putting it All Together



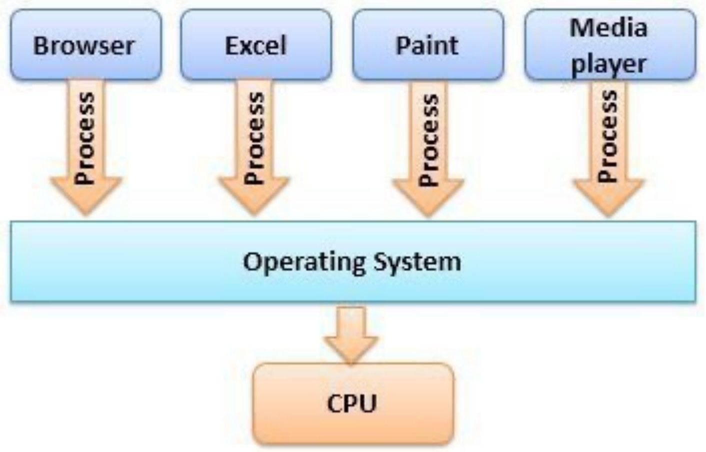
Operating System Structure

- **Multiprogramming (Batch system)**
  - Needed for efficiency
  - Single process cannot keep CPU and I/O devices busy at all times
  - Organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job is selected and run via **job scheduling**
  - When it has to wait for I/O, OS switches to another job



Operating System Structure

- **Timesharing (multitasking):** CPU switches jobs so frequently that users can interact with each job while it is running
- **interactive** computing
  - User interaction via input devices
  - **Response time** should be minimal
  - Each user has at least one program executing in memory ⇨ **process**
  - If several processes ready to run at the same time ⇨ **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes larger than physical memory



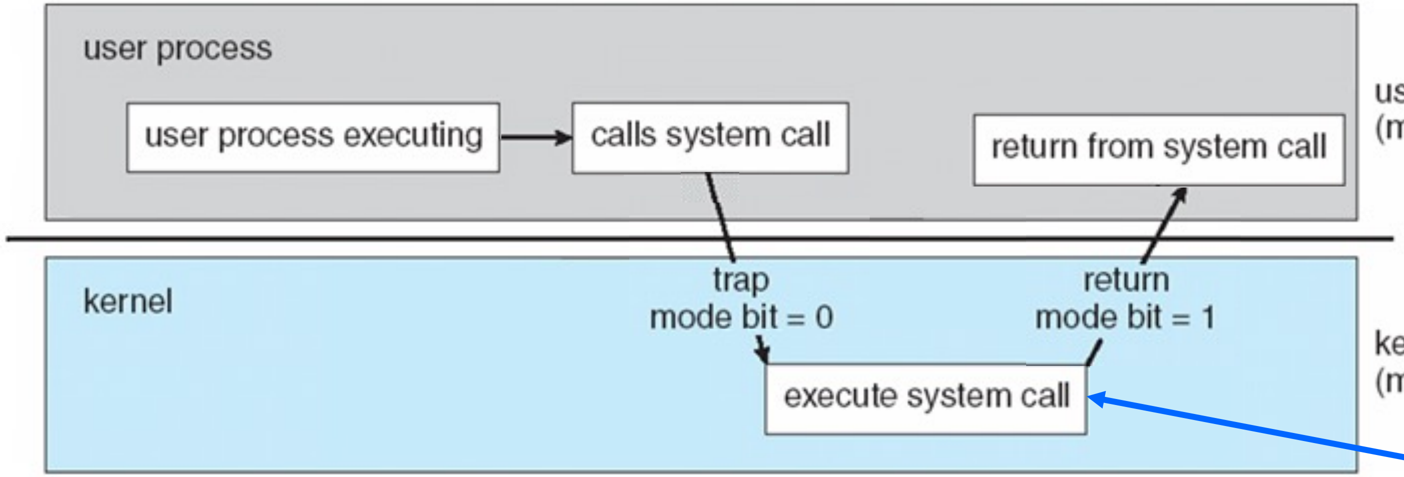
Multitasking

Operating System Operations

- **Interrupt driven** - hardware and software
- Hardware interrupt by one of the devices
- Software interrupt (**exception** or **trap):**
    - Software error (e.g., division by zero, invalid memory access)
    - Request for operating system service
    - Other process problems include infinite loop, processes modifying each other or the operating system

Operating System Operations

- **Dual-mode** operation allows OS to protect itself and protect users from one another
- **User mode** and **Kernel/Supervisor/System/Privileged mode**
- **Mode bit** provided by hardware
  - Provides ability to distinguish when system is running user code or kernel code
  - Some instructions designated as **privileged**, only executable in kernel mode
  - **System call** changes mode to kernel, return from call resets it to user



locate ISR via interrupt vector and excecute

Operating System Operations

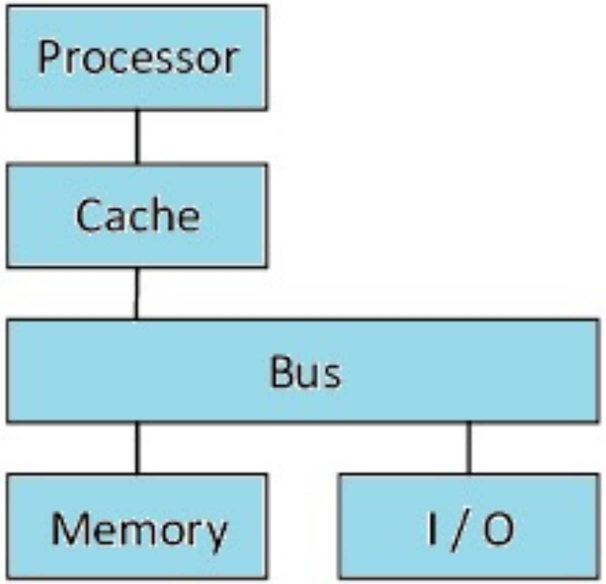
- Boot time → hardware starts in kernel mode
- After loading OS, user applications are started in user mode
- When trap/interrupt occurs, hardware switches from user mode to kernel mode
- examples of privileged instructions – switch to kernel mode, I/O control, timer management, interrupt management

Operating System Operations: Timer

- User processes must return control to OS
- Prevent infinite loop / process hogging resources
- Set to interrupt the computer after some time period
- Keep a counter that is decremented for every physical clock tick
- Operating system sets the counter (privileged instruction) before switching to user mode
- When counter reaches zero, generate an interrupt
- Set up before scheduling process to regain control or terminate program that exceeds allotted time

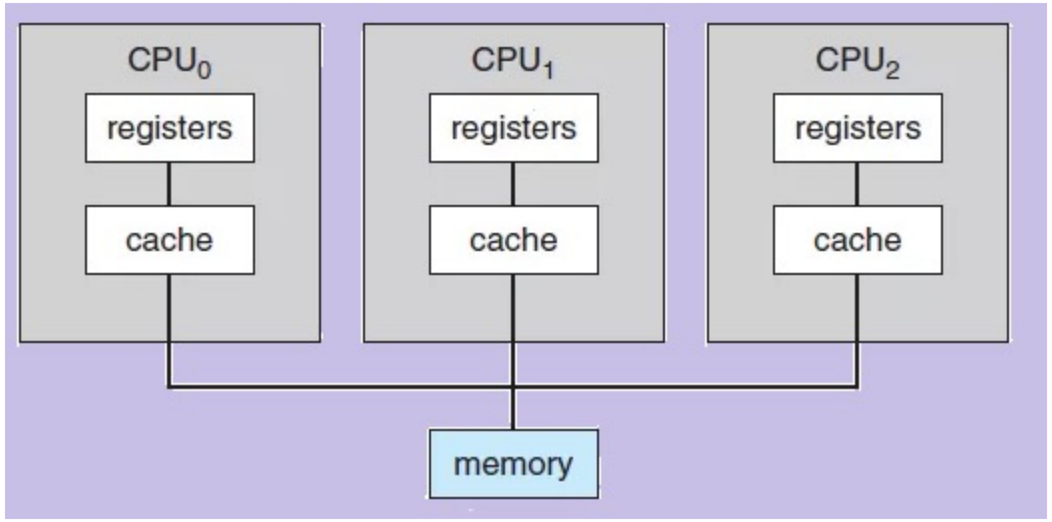
Computing Environments

Single-Processor Systems - one main CPU executing instructions, including instructions from user processes, some device-specific processors like disk, keyboard and graphics controller and I/O processor may be present



Computing Environments

- **Multiprocessors**
  - Also known as **parallel systems**, **multi-core systems**
  - 2 or more processors in close communication, sharing the computer bus and sometimes the clock, memory and peripheral devices
  - Advantages:
    - **Increased throughput**
    - **Economy of scale**
    - **Increased reliability** – graceful degradation, fault tolerance

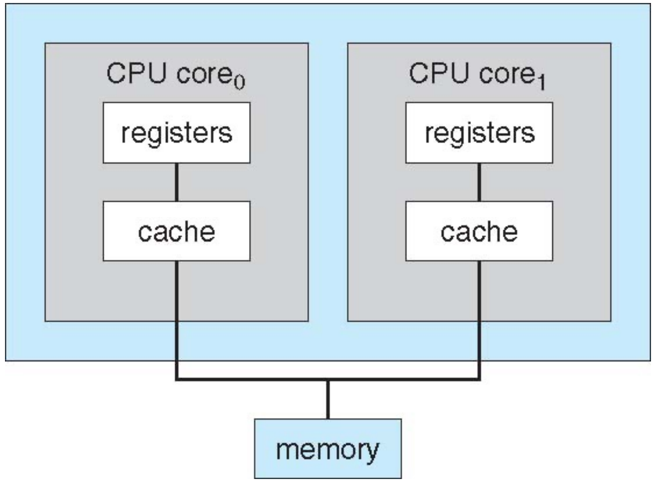


- Two types:
  - **Asymmetric Multiprocessing** – each processor is assigned a specific task, boss processor controls worker processors
  - **Symmetric Multiprocessing** – each processor performs all tasks, peers

Computing Environments

❖ Multicore Systems

- ❖ include multiple computing cores on a single chip
- ❖ more efficient than multiple chips with single cores because on-chip communication is faster than between-chip communication

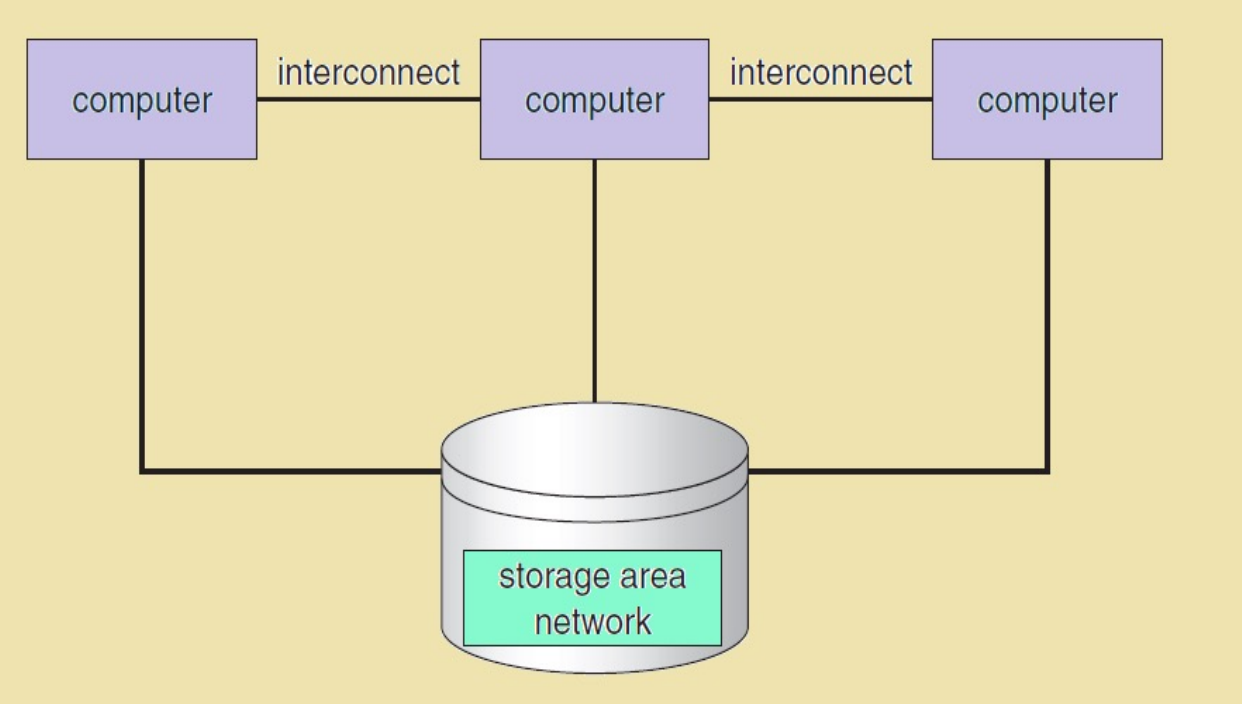


dual-core design with both cores on same chip

Computing Environments

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, users can see only a brief interruption of service
  - **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**

Computing Environments

Traditional Computing

- stand-alone general purpose machines
- most systems interconnect with others (i.e., the Internet)
- mobile computers interconnect via wireless networks
- home computers

Computing Environments

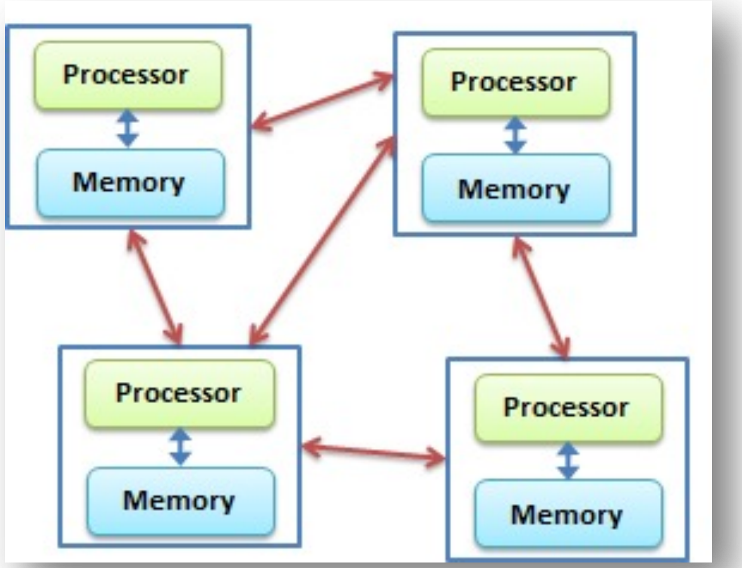
Mobile Computing

- handheld smartphones, tablets, etc.
- portable, lightweight
- allows different types of apps
- use wireless, or cellular data networks for connectivity
- Apple iOS, Google Android

Computing Environments

Distributed Computing

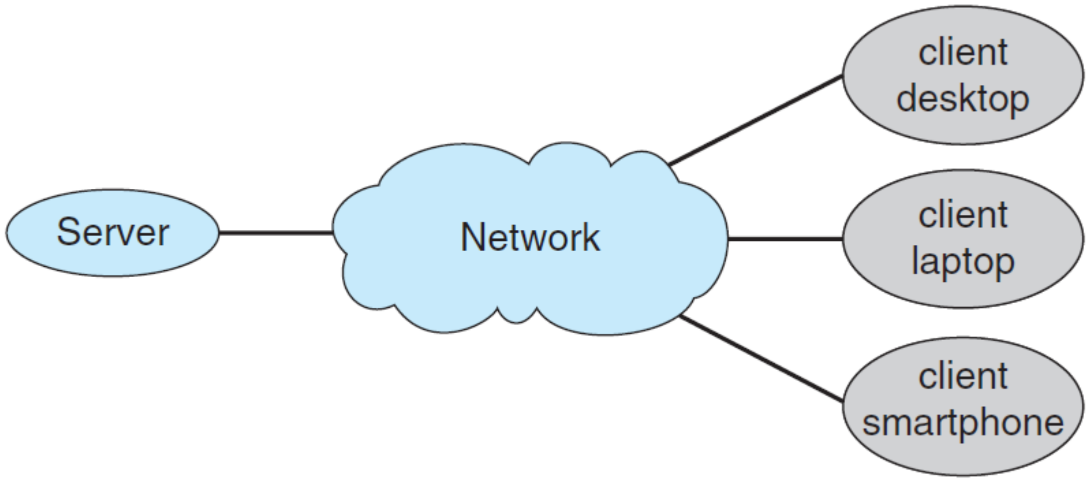
- collection of separate, possibly heterogeneous, systems networked together
- access to shared resources
- network is a communications path
  - Local Area Network (LAN)
  - Wide Area Network (WAN)
  - Metropolitan Area Network (MAN)
  - Personal Area Network (PAN)
- systems exchange messages



Computing Environments

Client Server Computing

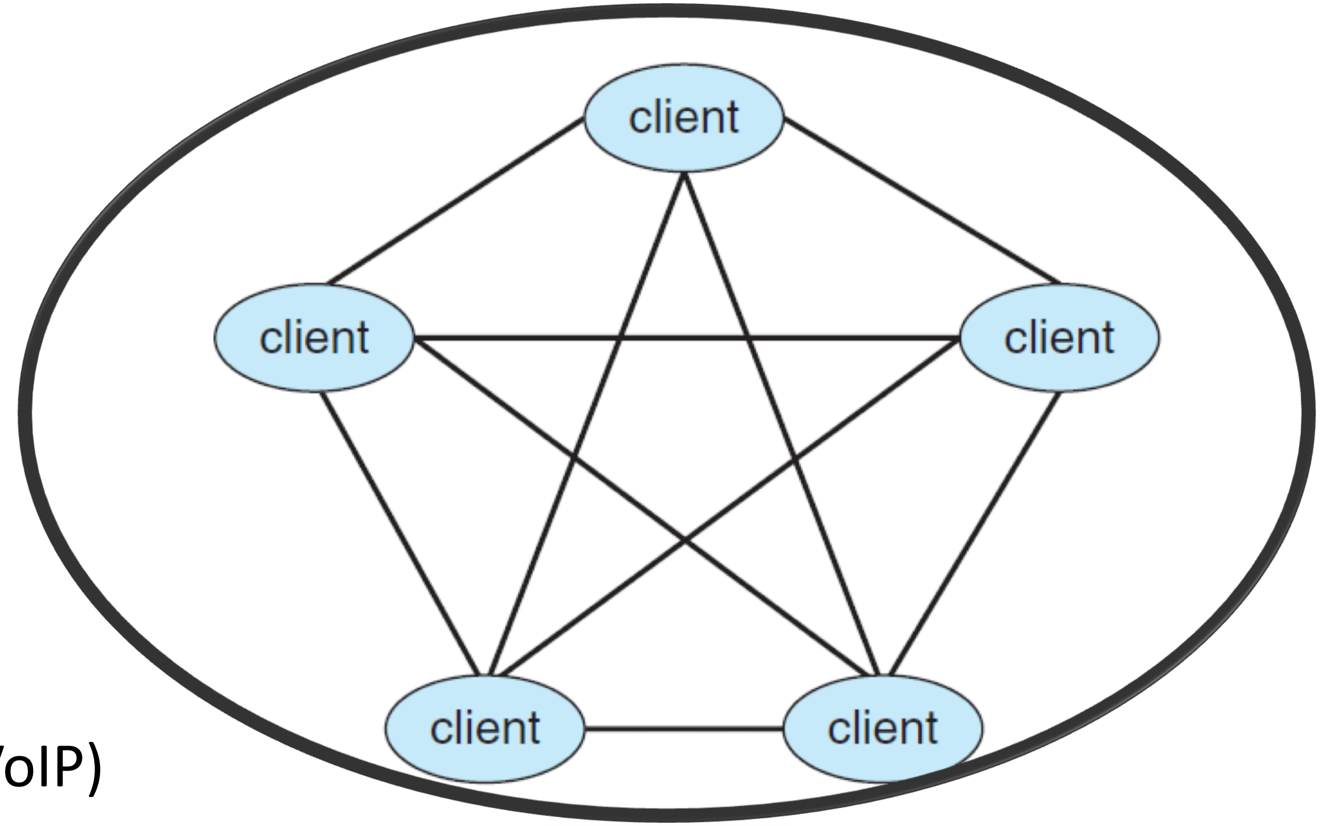
- terminals are PCs and mobile devices
- servers respond to requests generated by clients
- servers can be of 2 types
  - **compute-server** system provides an interface to client to request services, server executes the action and sends the results to clients
  - **file-server system** provides interface for clients to create, update, read and delete files



Computing Environments

Peer-to-peer Computing

- ❖ does not distinguish clients and servers
- ❖ nodes join and may also leave P2P network
- ❖ advantage over client server system



- ❖ Napster, Gnutella, BitTorrent, Skype (VoIP)

Thank You