Operating Systems:
Lecture 1

Introduction

Jinwoo Kim

jwkim@jjay.cuny.edu

Chapter 1: Introduction

- What is an Operating System?
- Mainframe Systems
- Desktop Systems
- Multiprocessor Systems
- Distributed Systems
- Clustered System
- Real-Time Systems
- Handheld Systems
- Computing Environments

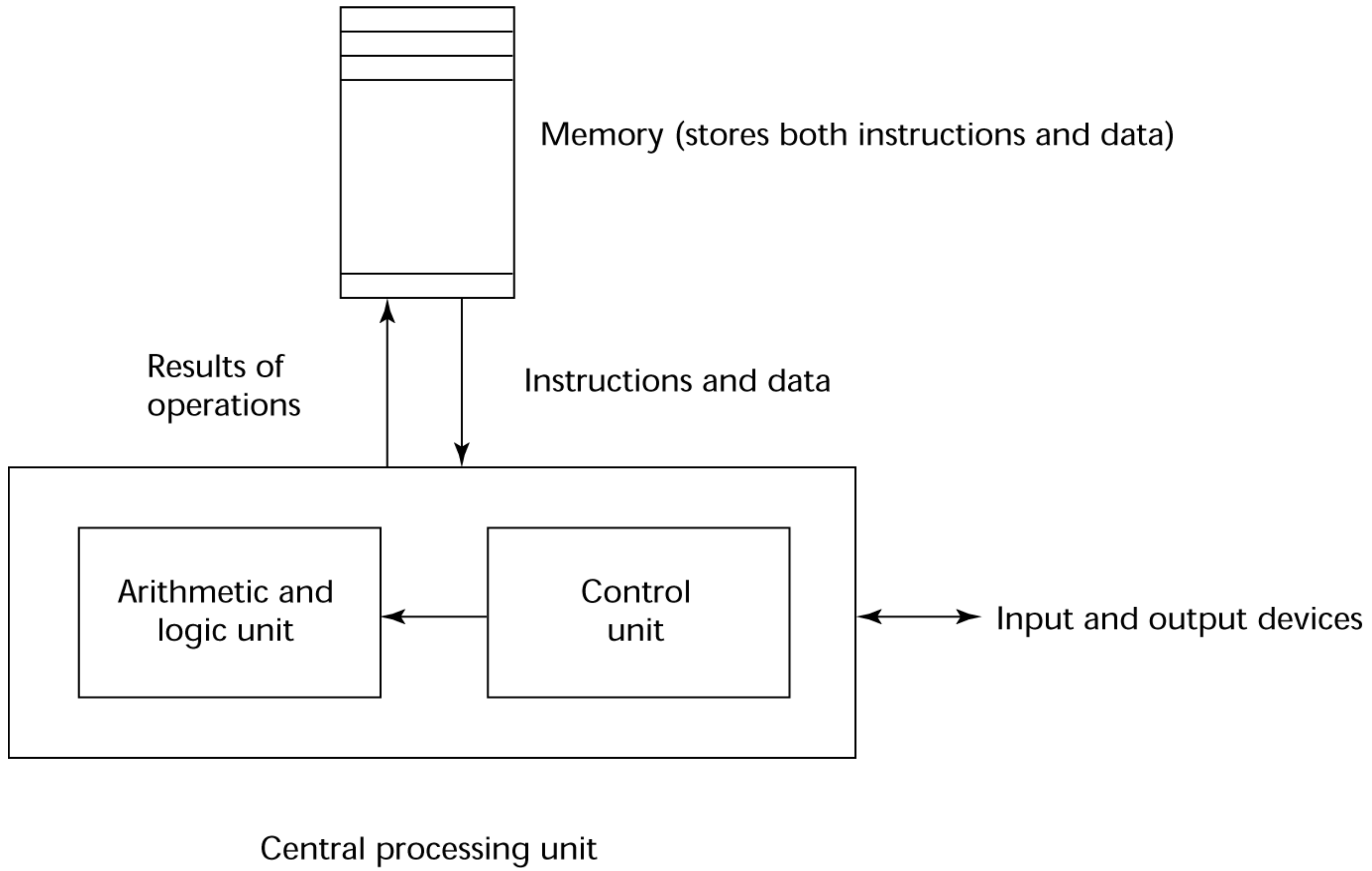
Basics

Fundamentals of Computer Systems and the Internet

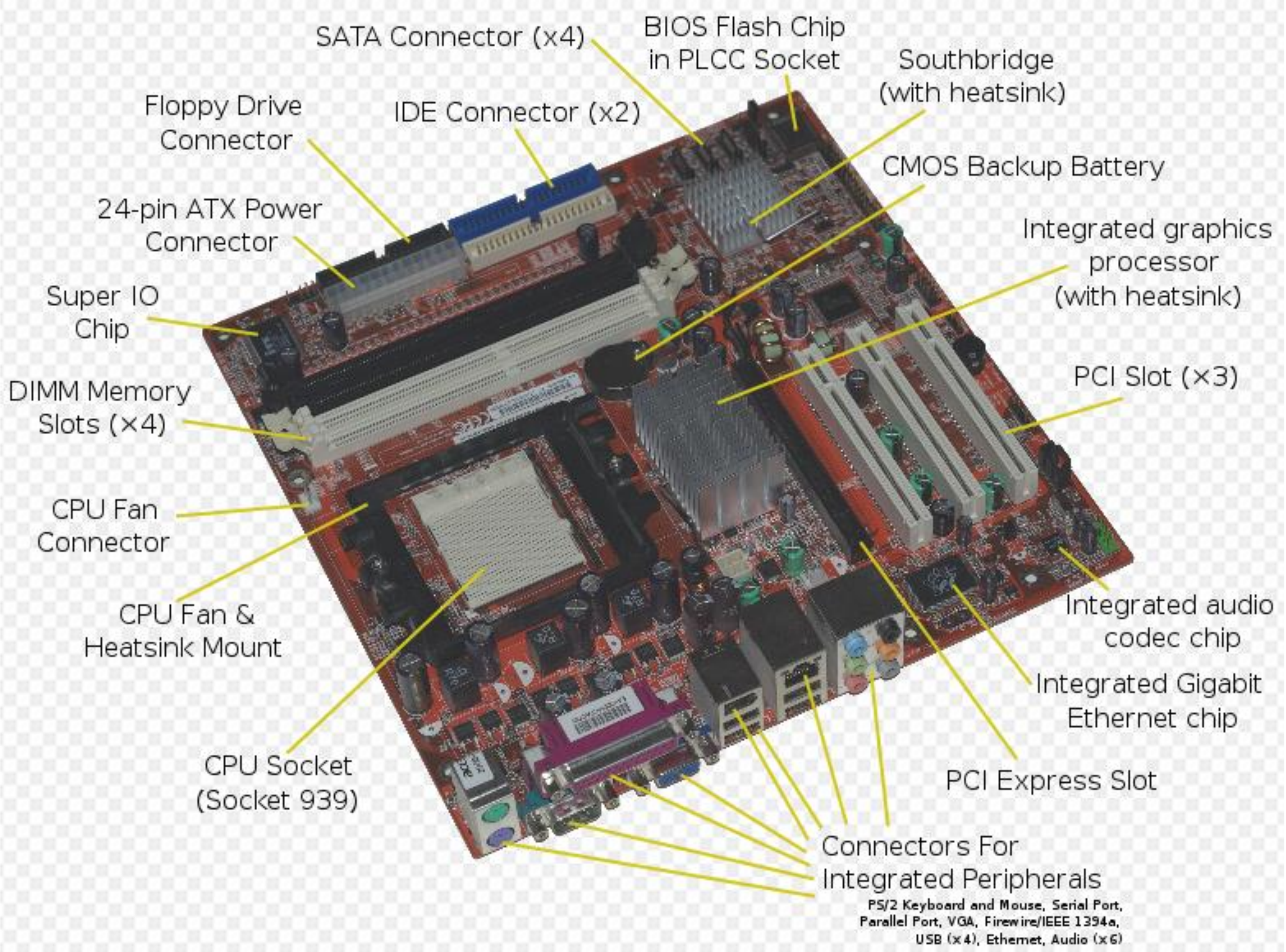
Is a computer really smart thing?

- Absolutely not!
  - Understands only 0 and 1's
  - Power of computing comes from its fast execution of rather primitive computations
- von Neumann Architecture
  - Most well-known and widely used computer architecture
  - Data and programs stored in memory
  - Memory is separate from CPU
  - Instructions and data are piped from memory to CPU
  - Basis for imperative languages
    - **Variables model memory cells**
    - **Assignment statements model piping**
    - **Iteration is efficient**

The von Neumann Architecture



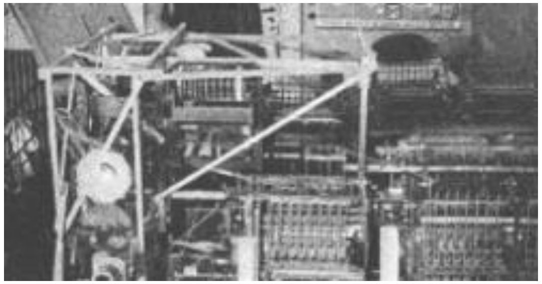
Example: Acer E360 Socket 939 motherboard



History of Computers

- Who invented computers?
  - Contribution from many inventors
  - A computer is a complex piece of machinery made up of many parts, each of which can be considered a separate invention.

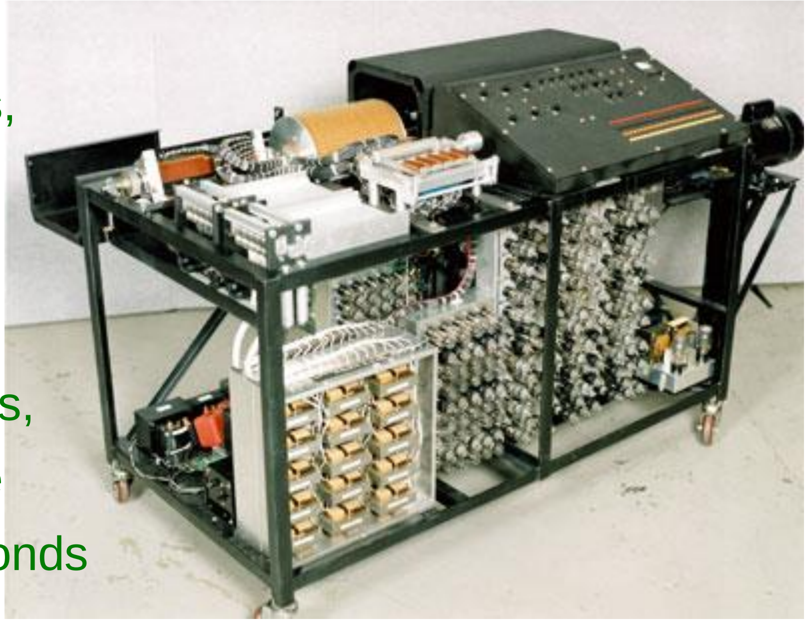
- In 1936, Konrad Zuse made a mechanical calculator using three basic elements: a control, a memory, and a calculator for the arithmetic and called it Z1, the first binary computer
  - First freely programmable computer
  - Konrad Zuse wrote the first algorithmic programming language called 'Plankalkül' in 1946, which he used to program his computers
  - He wrote the world's first chess-playing program using Plankalkül



Konrad Zuse's Z1
Circa 1938

History of Computers (Continued)

- Professor John Atanasoff and his graduate student Clifford Berry built the world's first electronic-digital computer at Iowa State University between 1939 and 1942
  - ABC computer
  - Several innovations in computing, including a binary system of arithmetic, parallel processing, and a separation of memory and computing functions
  - The final product was the size of a desk, weighed 700 pounds, had over 300 vacuum tubes, and contained a mile of wire
  - It could calculate about one operation every 15 seconds, today a computer can calculate 150 billion operations in 15 seconds



History of Computers (Continued)

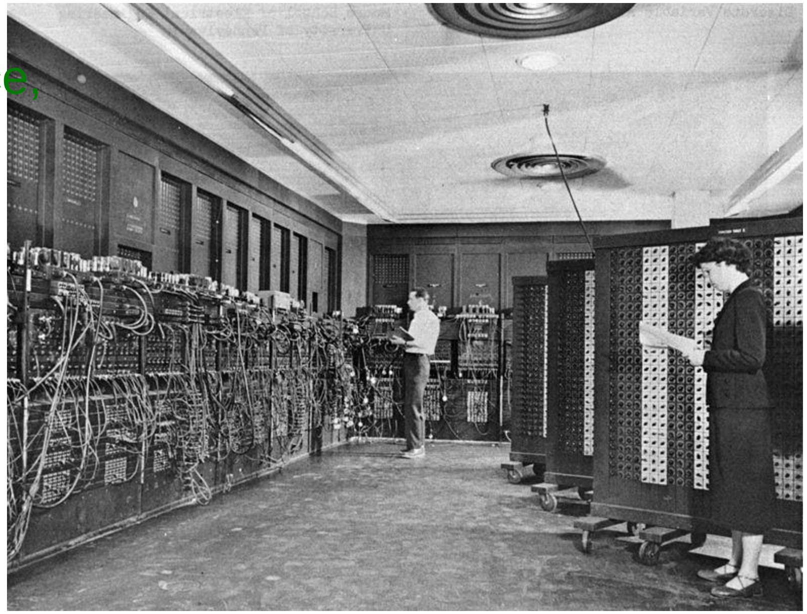
- In 1946, John Mauchly and J Presper Eckert developed the ENIAC I (**E**lectrical **N**umerical **I**ntegrator **A**nd **C**alculator)
  - By support from U.S. military and considered as first modern computer
  - The ENIAC contained 17,468 vacuum tubes, along with 70,000 resistors, 10,000 capacitors, 1,500 relays, 6,000 manual switches and 5 million soldered joints
  - It covered 1800 square feet

    (167 square meters) of floor space,

    weighed 30 tons,

    consumed 160 kilowatts of

    electrical power



History of Computers (Continued)

- In 1948, "The Transistor" was first built by John Bardeen, Walter Brattain & William Shockley
  - Semiconductor device used to amplify and switch electronic signals
  - The transistor is the key active component in practically all modern electronics, and is considered by many to be one of the greatest inventions of the twentieth century

History of Computers (Continued)

- In 1951, John Presper Eckert & John W. Mauchly built first commercial computer called “UNIVAC”
  - By doing the research for their customer, United States Census Bureau
    - **The Bureau needed a new computer to deal with the exploding U.S. population (the beginning of the famous baby boom)**
  - Able to pick presidential winners (Eisenhower vs. Stevenson)

- In 1953, IBM enters into “The History of Computers” with IBM 701 EDPM Computer
  - In 1954, the first successful high level programming language, Fortran, was developed by John Backus & IBM

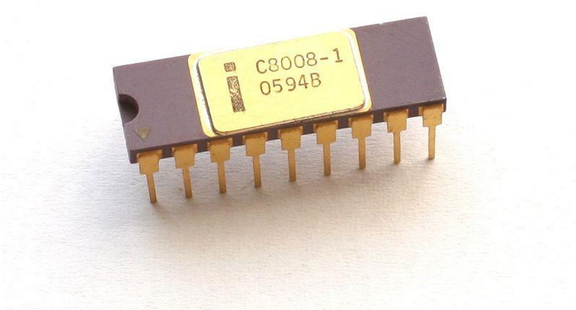
History of Computers (Continued)

- In 1958, The Integrated Circuit, otherwise known as “The Chip”, was developed by Jack Kilby & Robert Noyce

- In 1964, IBM unveiled first “mainframe computer” with the System/360
  - Cost to develop: $5 billion ($30 billion in today's dollars)
    - **But the gamble paid off: company's revenue jumped from $3.2 billion the year it was introduced to $7.5 billion in 1970**
  - Major breakthrough in the technology and business worlds
  - Allowed companies to perform multiple tasks at the same time on a single machine
    - **Before then, a user would have to schedule time on the company computer to do a specific task, whether to process payroll or analyze business expenses**
  - Dominated computing industry until PC revolution in the 1980s

History of Computers (Continued)

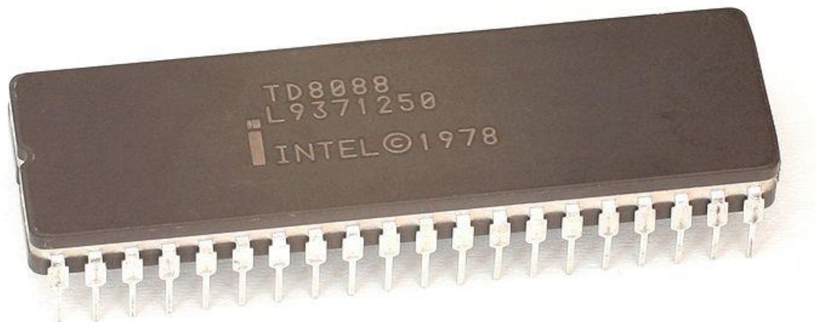
- In 1969, “ARPAnet”, the origin of the Internet, was constructed
  - Packet-switching development
  - ARPA introduces network for defense and develops e-mail and US universities join network in 1970
- In 1971, Faggin, Hoff & Mazor made “Intel 4004 Computer Microprocessor”
  - The first microprocessor
- In 1972, Intel introduces world’s first 8-bit microprocessor, Intel 8008



History of Computers (Continued)

- In 1978, Intel introduces era of ***x86*** with 16-bit 8086 CPU

- In 1979, variant of 8086 microporcessor, “Intel 8088”, was introduced and chosen later as the brain of first personal computer by IBM



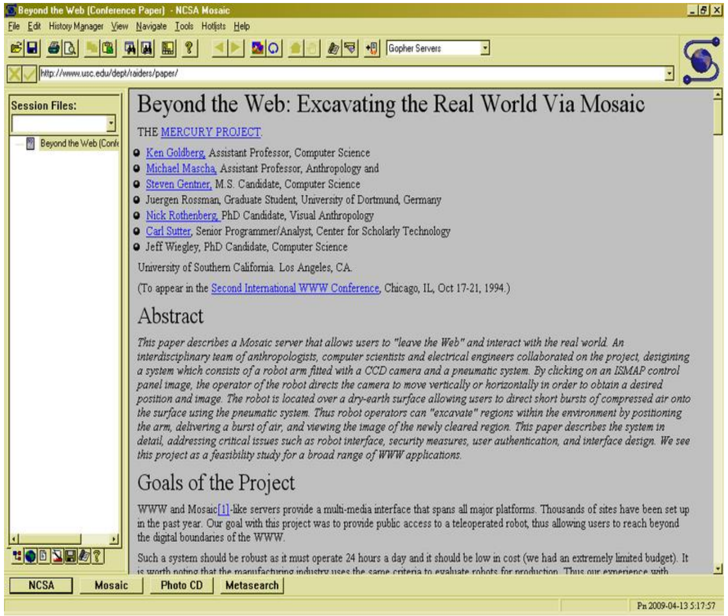
- In 1981, IBM introduced
  “IBM PC - Home Computer”
  and Microsoft revealed
  its “MS-DOS” Operating System

History of Computers (Continued)

- In 1983, The first home computer with a GUI, graphical user interface, was developed by Apple
- In 1985, Microsoft begins the friendly war with Apple with its launch of Microsoft Windows operating system
- By the early 1990s, sales of mainframes, then IBM's main product, were dropping dramatically in the face of stiff competition from rivals such as Sun Microsystems
  - Also, instead of big boxes in the back room, companies turned to servers that connected PCs in a network
  - People predicted it will extinct in decades

History of Computers (Continued)

- In 1993, “Mosaic” by Marc Andreessen, the first properly developed web-browser, takes Internet by storm



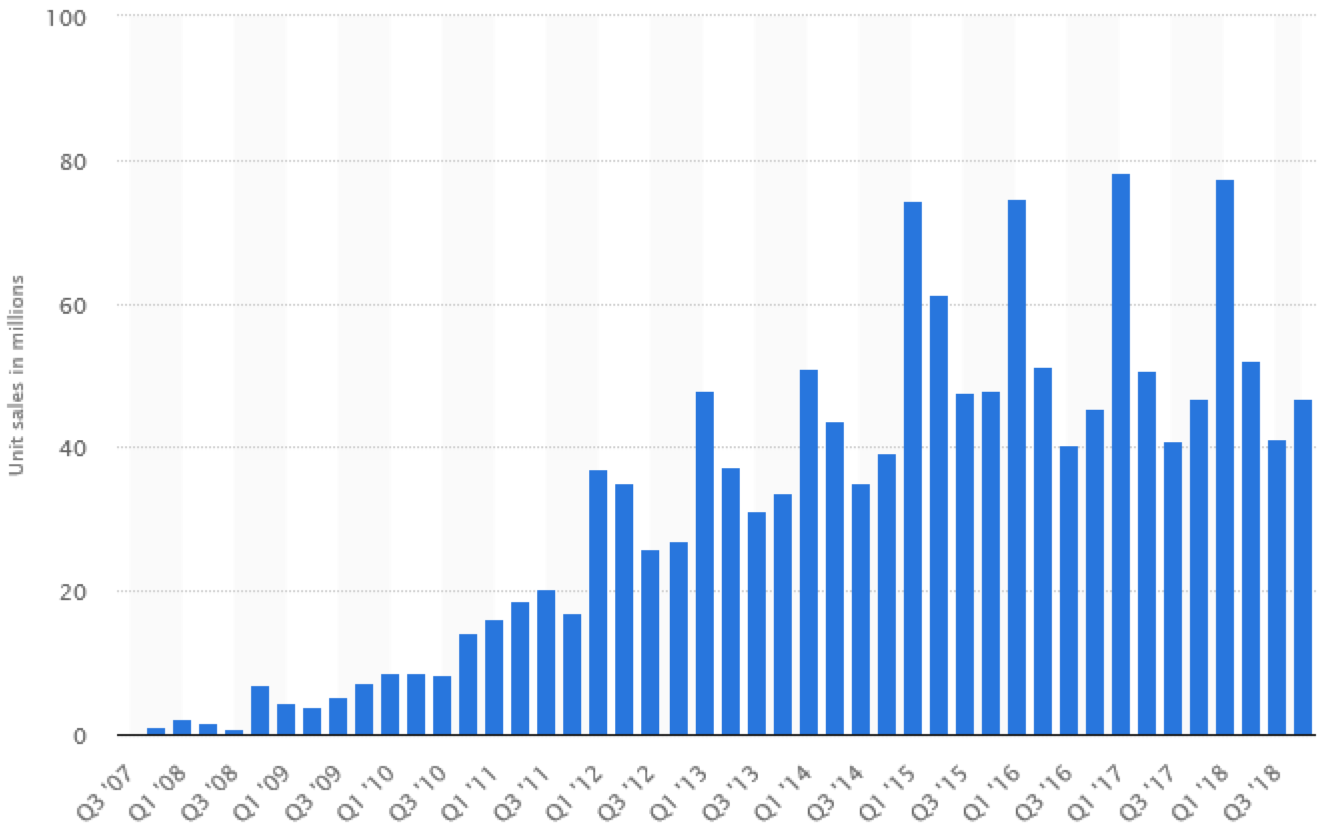
- In 1994, Netscape changed the picture of computing industry
  - Finally the era of the Internet arrived
  - In a 2007 *PC World* column, the original Netscape Navigator was considered the "best tech product of all time" due to its impact on the Internet

History of Computers (Continued)

- 1995-2005 : Internet and the 32-bit era
  - MS released Windows 95 on 8/24/1995
    - **Came with pre-emptive multitasking, new user interface, Explorer web browser**
  - Windows XP on 2001
    - **$613 million fine from EU antitrust legal action to produce new version of XP without WMP**

History of Computers (Continued)

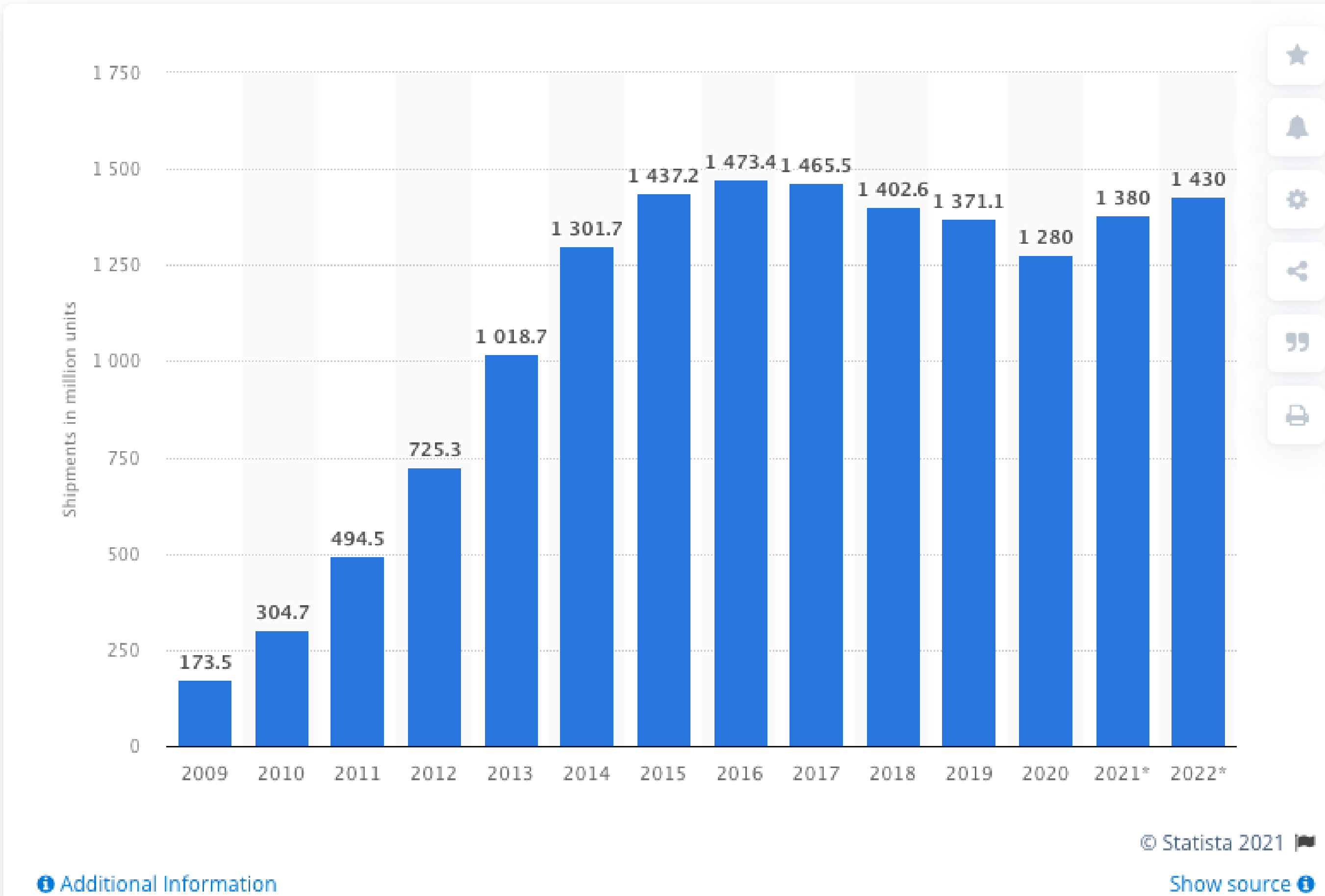
- 2007 ~ 2024: Smart devices and Cloud computing
  - Apple's iPhone introduced in 2007 changed the picture of computing industry (almost 90 million iPhones sold within 3 and ½ years)



History of Computers (Continued)

Global smartphone shipments forecast from 2010 to 2022

(in million units)



History of Computers (Continued)

- We live in the age of "embedded computer" world now
  - Downshift of "center of gravity of computing"
  - But at the same time, IBM sells record number of mainframe computers
  - Twelve years after its initial release, altogether about 1.5 billion iPhones have been shipped worldwide, making it one of the most used smartphones in the

- So, we have a history of computers about 70+ years!
  - Compared to 50+ years of Cyber world

What is an Operating System?

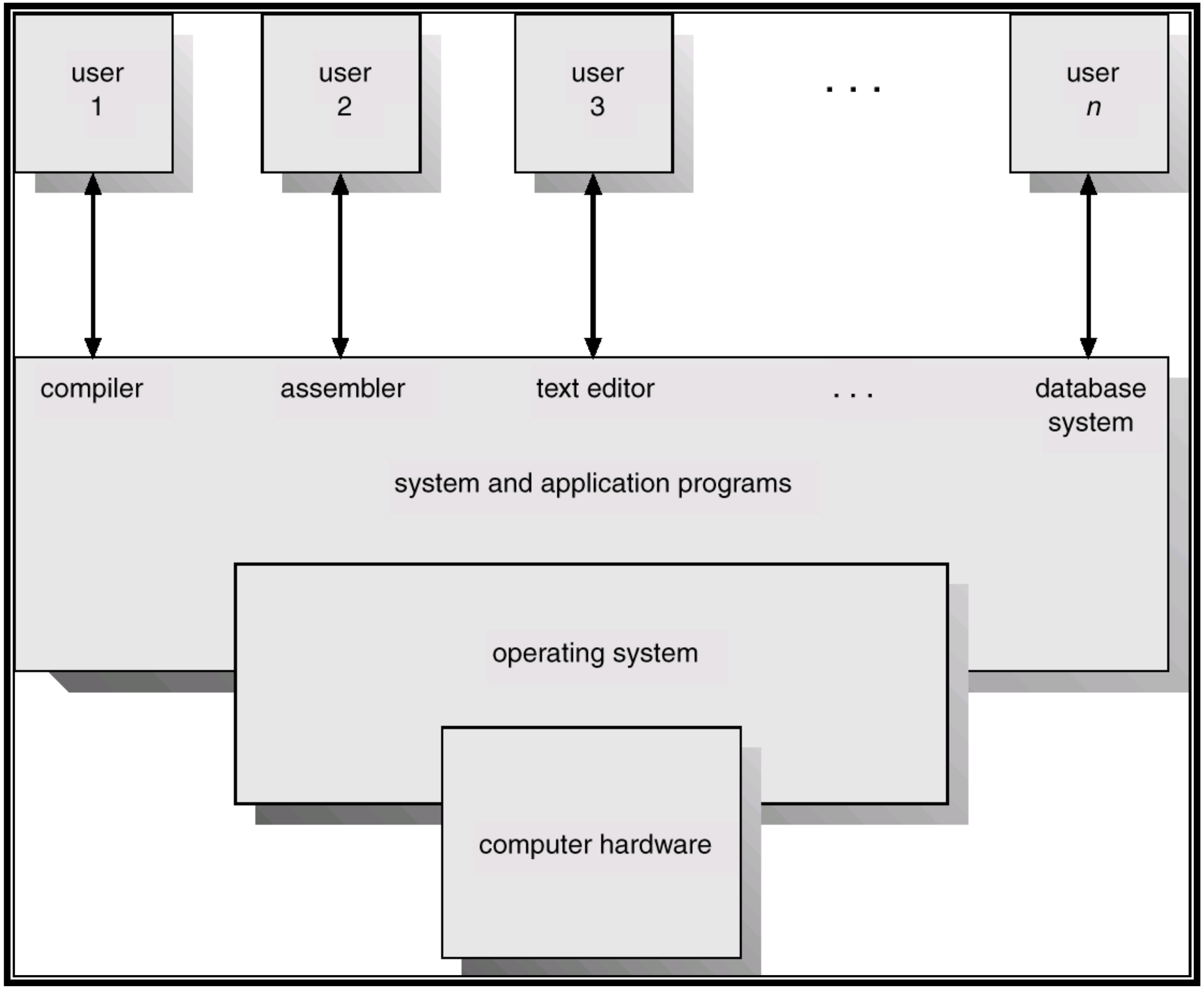
- A program that acts as an intermediary between a user of a computer (application software) and the computer hardware

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

Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, I/O devices)
2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users
3. Applications 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, business programs)
4. Users (people, machines, other computers)

Abstract View of System Components



What Operating Systems Do

- Depends on the point of view
- Users want convenience, **ease of use** and **good performance**
  - Don't care about **resource utilization**
- But shared computer such as **mainframe** or **minicomputer** must keep all users happy
- Users of dedicate 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

- Resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use

- Control program
  - Controls the execution of programs to prevent errors and improper use of the computer

Operating System Definition (Continued)

- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is a 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

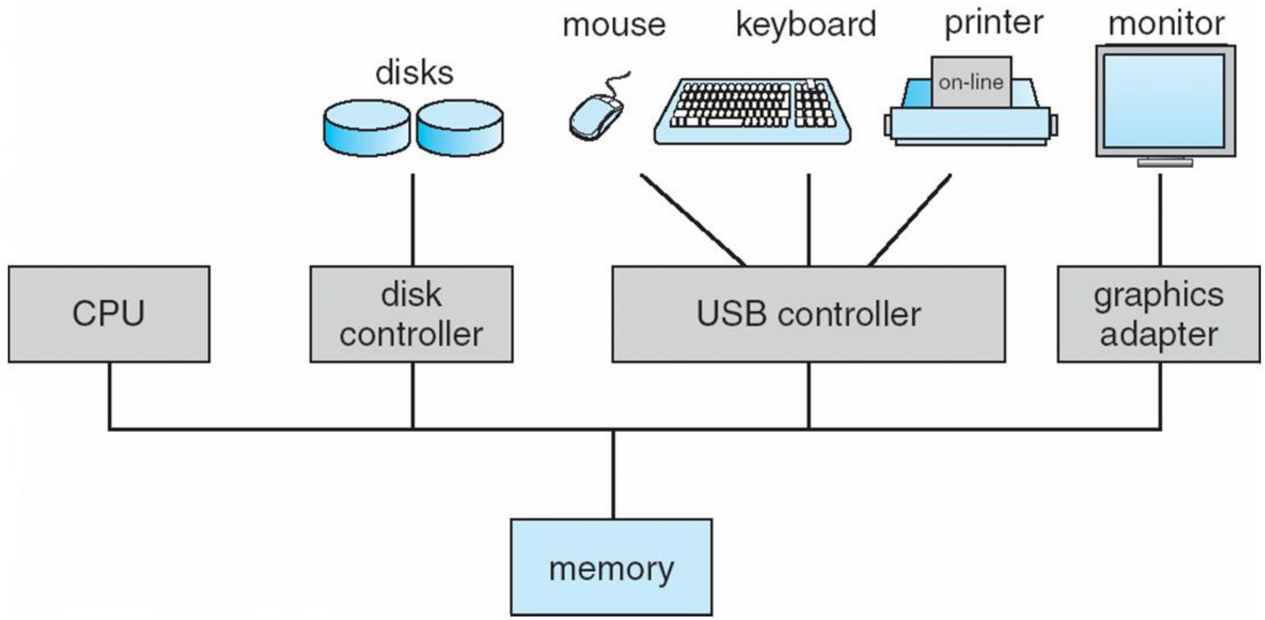
Computer Startup

- **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

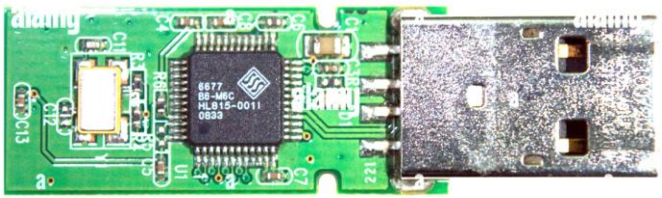
- Details later

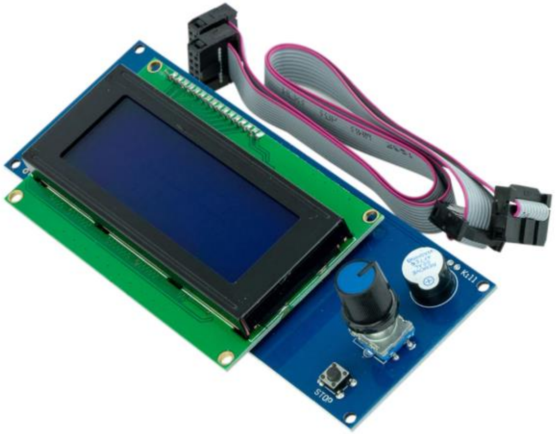
Computer System Organization

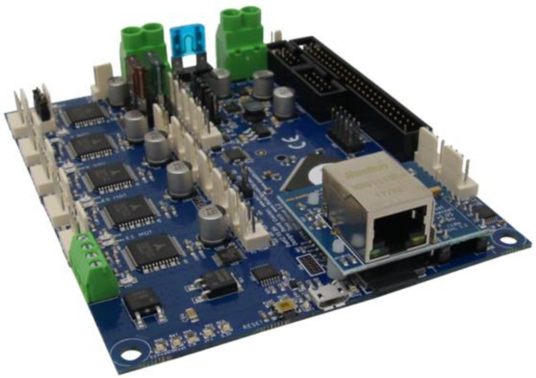
- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



Some Device Controllers







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

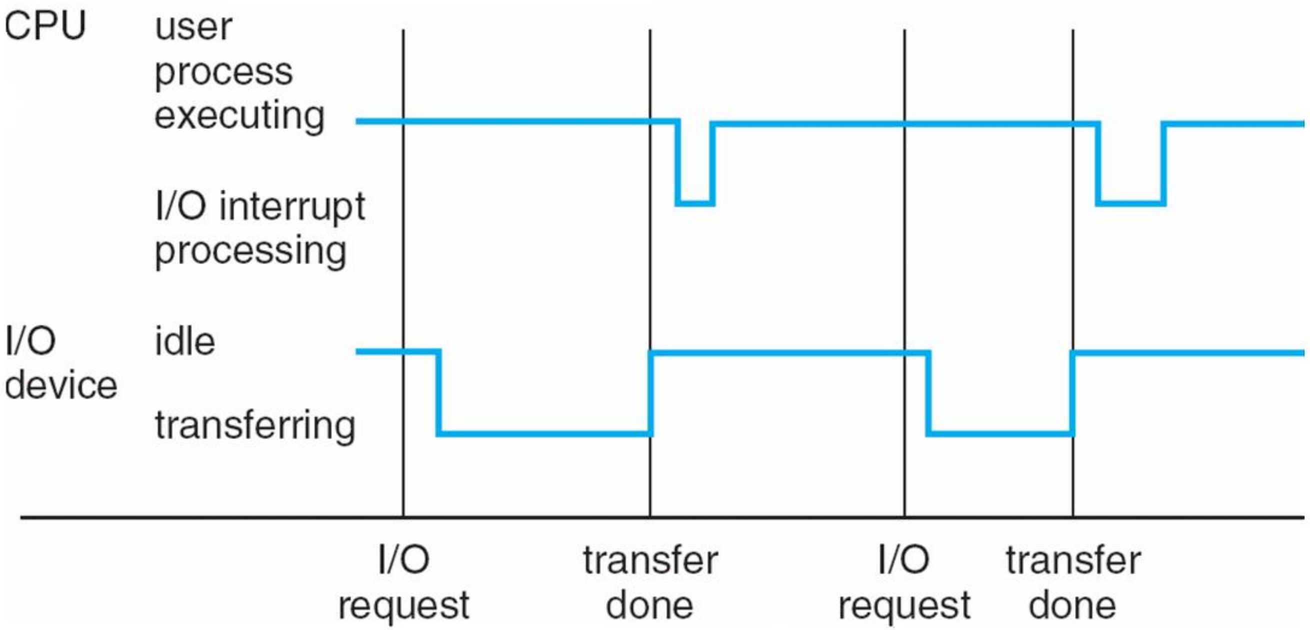
Common Functions of Interrupts

- 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
- A **trap** or **exception** 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
- Determines which type of interrupt has occurred:
  - **polling**
  - **vectored** interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt

Interrupt Timeline



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 OS 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
  - OS indexes into I/O device table to determine device status and to modify table entry to include interrupt

Storage Definitions and Notation Review

- Bit: Basic unit of computer storage
  - It can contain one of two values, “0 or 1”
  - All other storage in a computer is based on collection of bits
  - Given enough bits, it is amazing how many things a computer can represent
    - **Numbers, letters, images, movies, sounds, documents,…**

- Byte: Collection of 8 bits
  - Traditionally, we used one byte to store one character or symbol
  - Also used to represent size of disk or memory
  - KB (1,024 bytes), MB ($1{,}024^2$ bytes, 1,000,000 bytes), GB ($1{,}024^3$ bytes, 1,000,000,000 bytes) , TB ($1{,}024^4$bytes, 1,000,000,000,000 bytes), Petabytes (PB, $1{,}024^5$bytes, 1,000,000,000,000,000 bytes)

- Word
  - Given computer architecture’s native storage unit
  - Generally made up of one or more bytes
  - E.g. A computer may have instructions to move 64-bit (8-byte) words

Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - **Random access**
  - Typically **volatile**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hard 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
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular

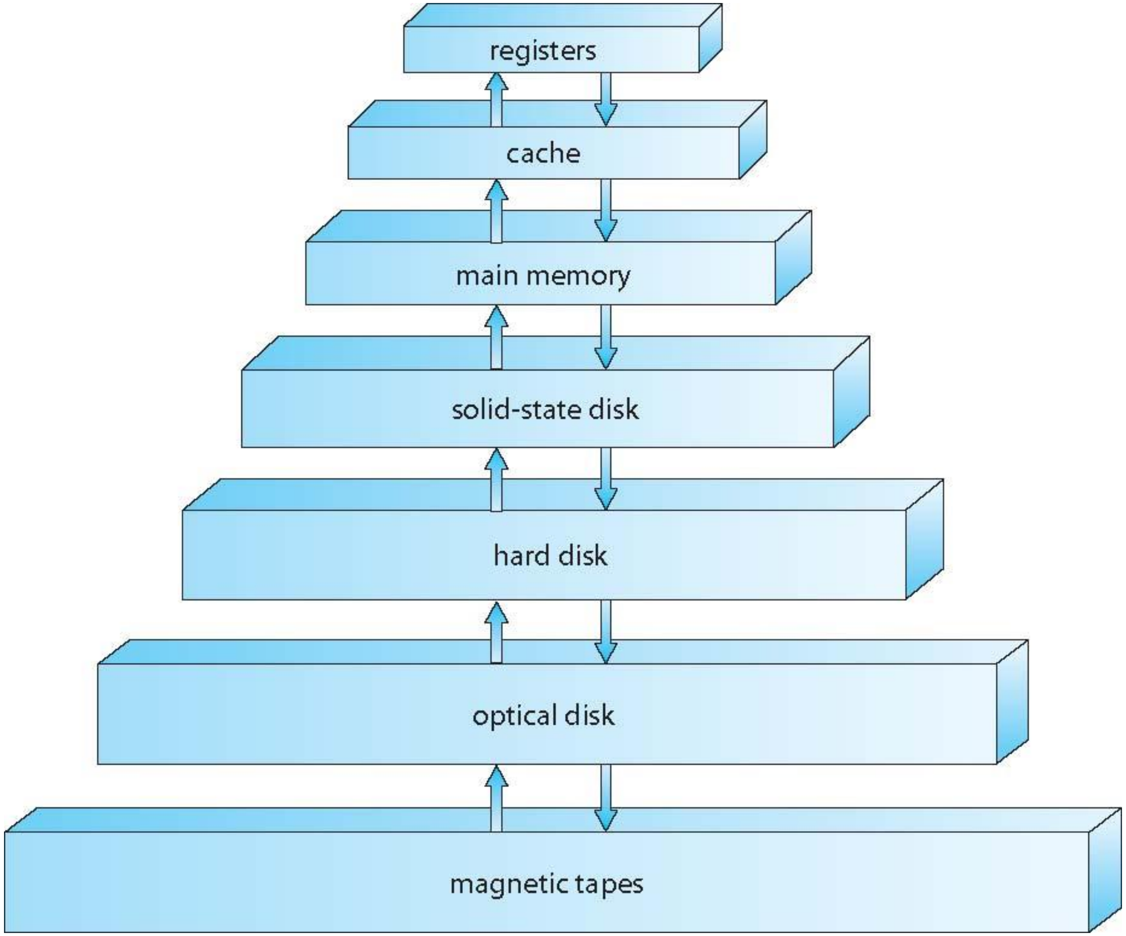
Storage Hierarchy

- 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

- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

Storage-Device Hierarchy



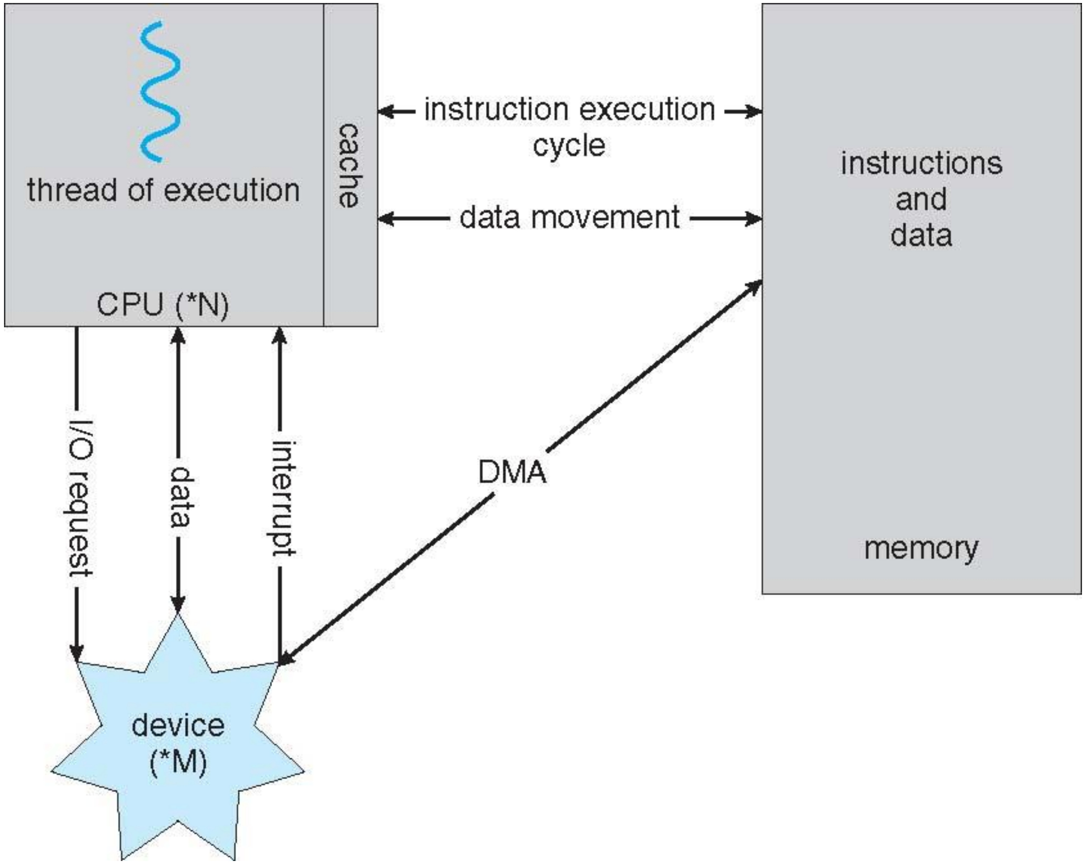
Caching

- 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

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

How a Modern Computer Works

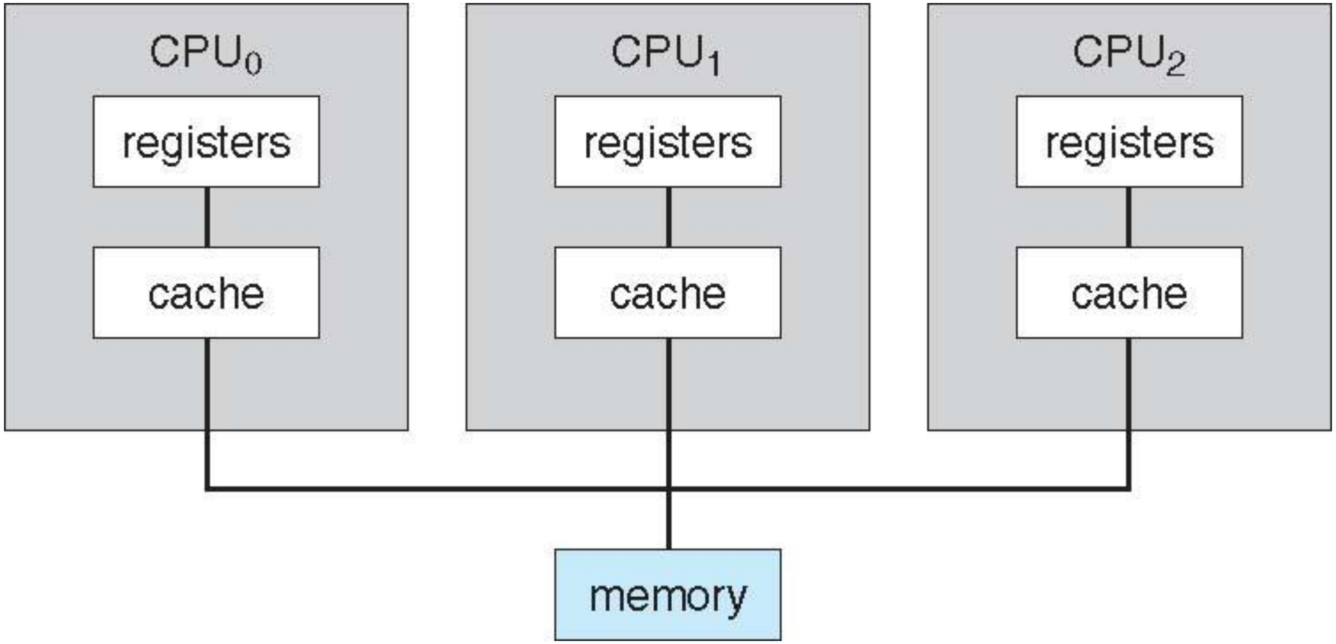


A von Neumann architecture

Computer-System Architecture

- Most systems use a single general-purpose processor
  - Most systems have special-purpose processors as well
- **Multiprocessors** systems growing in use and importance
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale**
    3. **Increased reliability – graceful degradation or fault tolerance**
  - Two types:
    1. **Asymmetric Multiprocessing – each processor is assigned a specie task.**
    2. **Symmetric Multiprocessing – each processor performs all tasks**

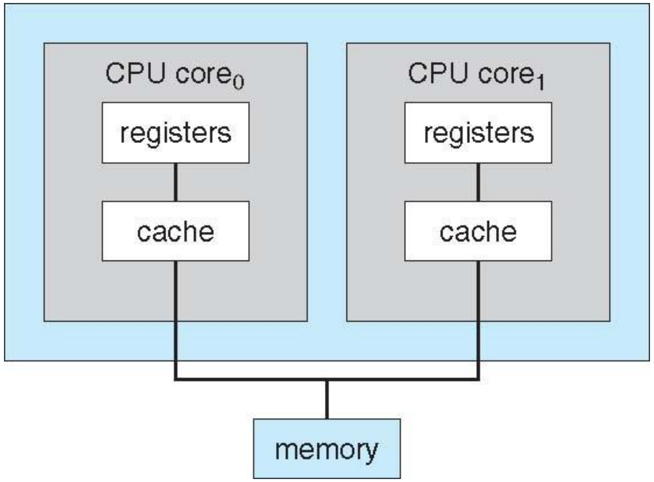
Symmetric Multiprocessing Architecture



A Dual-Core Design

- Multi-chip and **multicore**

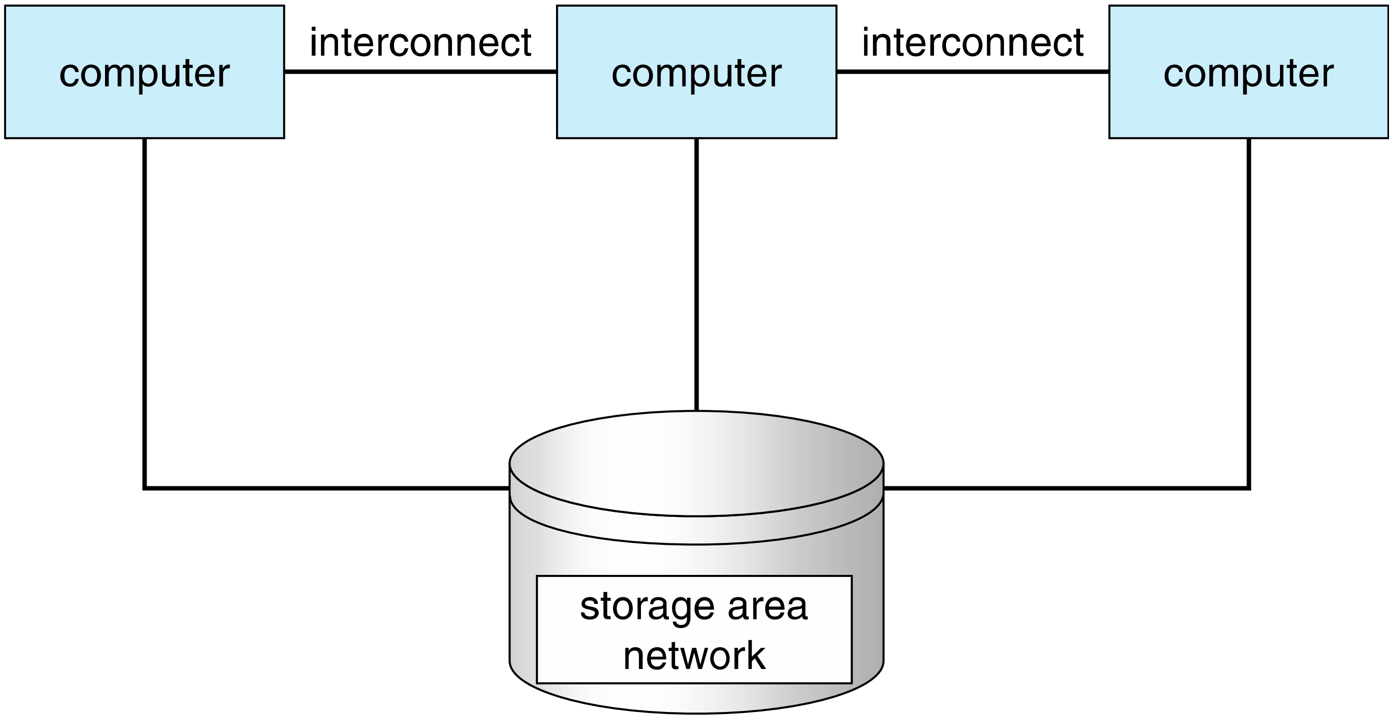
- Systems containing all chips
  - Chassis containing multiple separate systems



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**
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

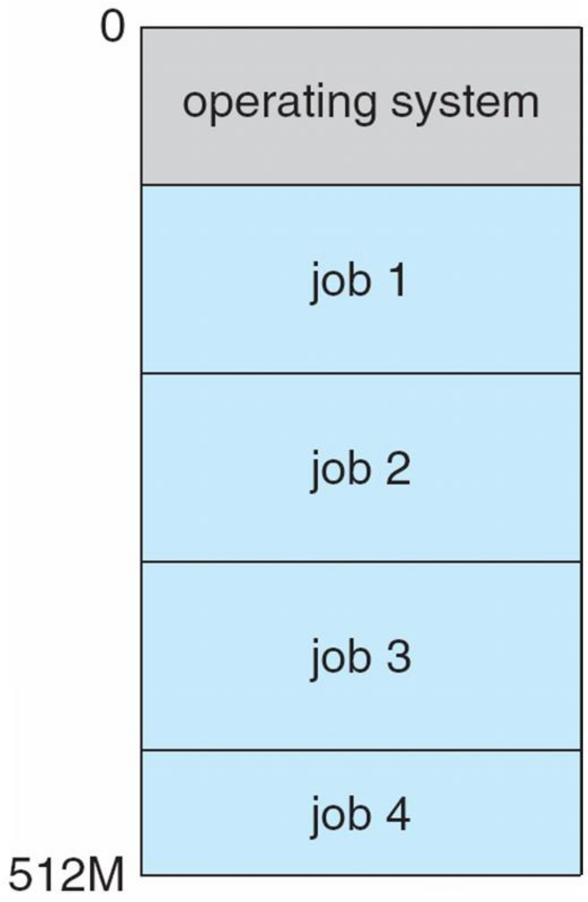
Clustered Systems (Continued)



Operating System Structure

- **Multiprogramming** (**Batch system**) needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming 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 selected and run via **job scheduling**
  - When it has to wait (for I/O for example), OS switches to another job
- **Timesharing** (**multitasking**) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing
  - **Response time** should be < 1 second
  - Each user has at least one program executing in memory ⇨**process**
  - If several jobs 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 not completely in memory

Memory Layout for Multiprogrammed System



Operating-System Operations

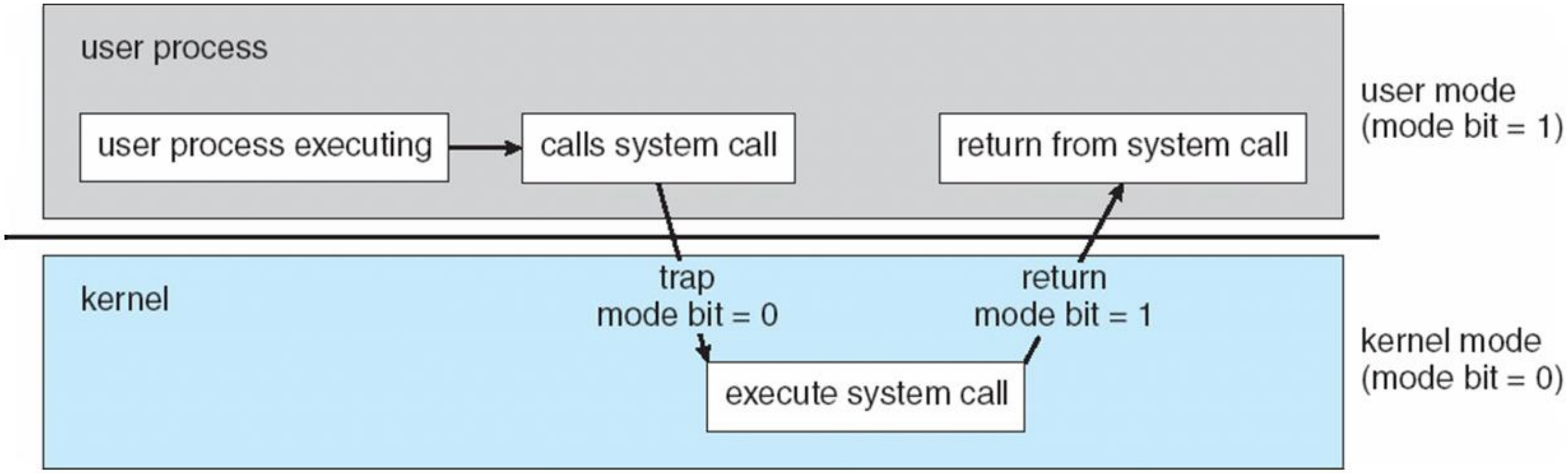
- **Interrupt driven** (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap):**
    - **Software error (e.g., division by zero)**
    - **Request for operating system service**
    - **Other process problems include infinite loop, processes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

Operating-System Operations (Continued)

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel 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**
- Increasingly CPUs support multi-mode operations
  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Timer is set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock.
  - Operating system set the counter (privileged instruction)
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a ***passive entity***, process is an ***active entity***.

- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data

- Process termination requires reclaim of any reusable resources
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

Process Management Activities

- The operating system is responsible for the following activities in connection with process management:
  - Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication
  - Providing mechanisms for deadlock handling

Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - **Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)**
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - **Creating and deleting files and directories**
    - **Primitives to manipulate files and directories**
    - **Mapping files onto secondary storage**
    - **Backup files onto stable (non-volatile) storage media**

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

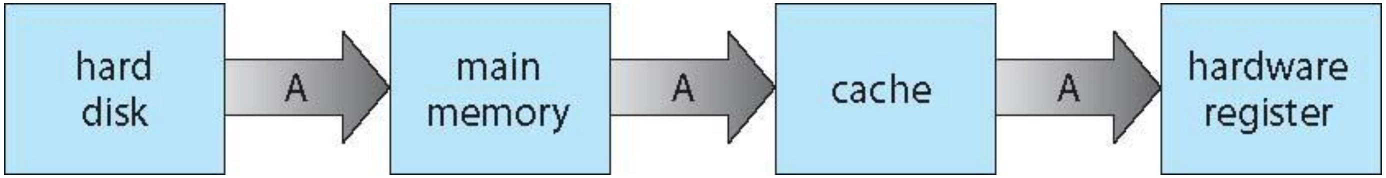
Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Movement between levels of storage hierarchy can be explicit or implicit

Migration of data "A" from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



- Multiprocessor environment must provide **cache coherency** in hardware such that all CPUs have the most recent value in their cache
- Distributed environment situation even more complex
  - Several copies of a datum can exist
  - Various solutions covered in Chapter 17

I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user

- I/O subsystem responsible for
  – Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  – General device-driver interface
  – Drivers for specific hardware devices

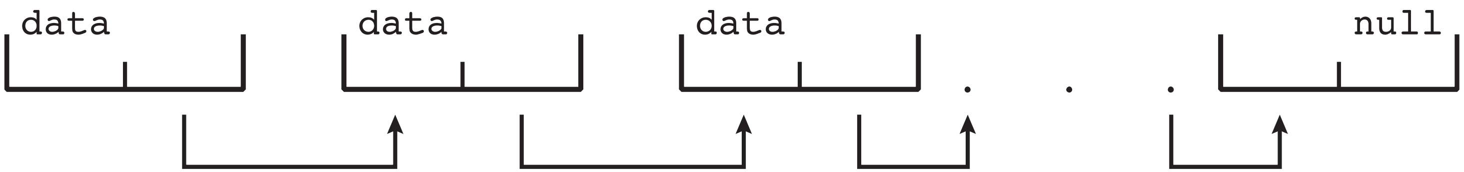
Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

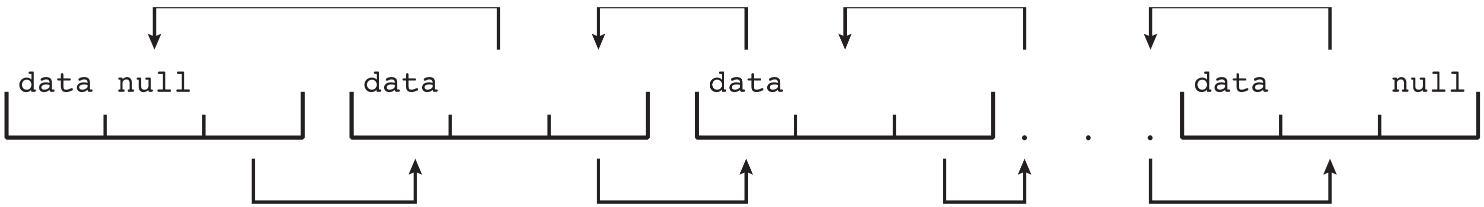
Kernel Data Structures

Many similar to standard programming data structures

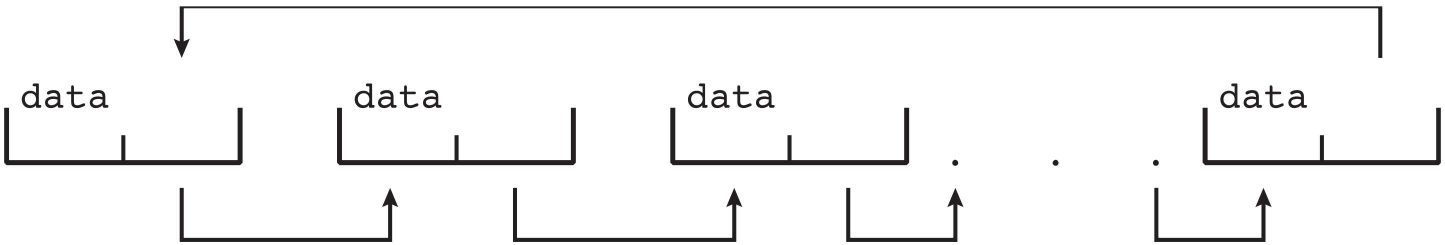
Singly linked list



Doubly linked list

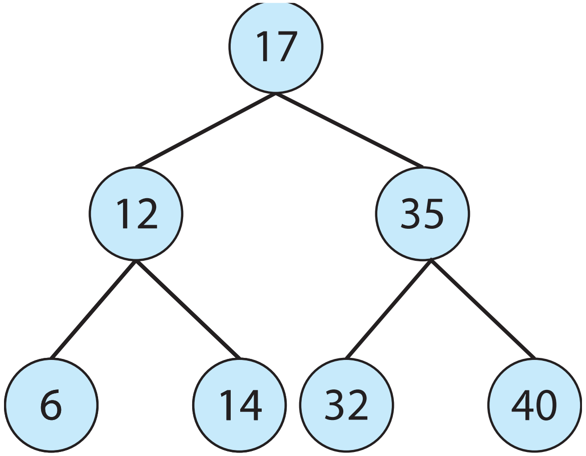


Circular linked list



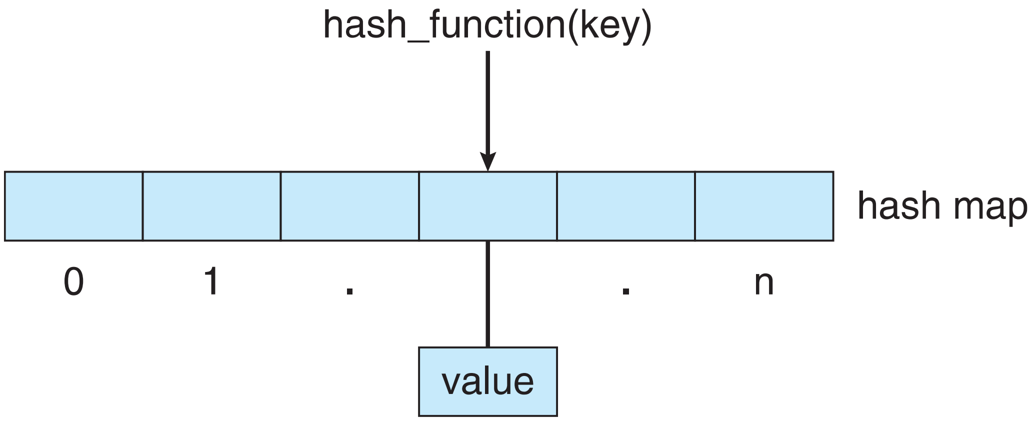
Kernel Data Structures

- **Binary search tree**
  left <= right
  - Search performance is *O(n)*
  - **Balanced binary search tree** is *O(lg n)*



Kernel Data Structures

- **Hash function** can create a **hash map**



- **Bitmap** – string of *n* binary digits representing the status of *n* items

- Linux data structures defined in
  include files `<linux/list.h>, <linux/kfifo.h>, <linux/rbtree.h>`

Computing Environments - Traditional

- Stand-alone general-purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers** (**thin clients**) are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks

Computing Environments - Mobile

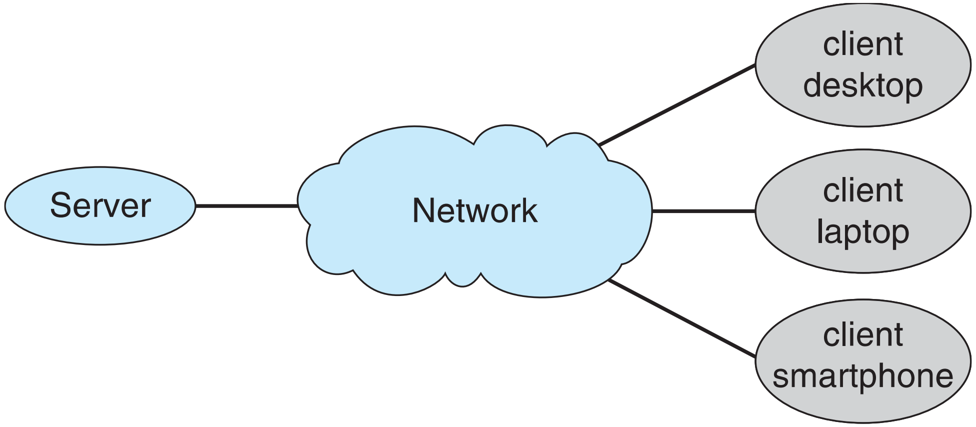
- Handheld smartphones, tablets, etc
- What is the functional difference between them and a “traditional” laptop?
- Extra feature – more OS features (GPS, gyroscope)
- Allows new types of apps like ***augmented reality***
- Use IEEE 802.11 wireless, or cellular data networks for connectivity
- Leaders are **Apple iOS** and **Google Android**

Computing Environments – Distributed

- Distributed computing
  - Collection of separate, possibly heterogeneous, systems networked together
    - **Network is a communications path, TCP/IP most common**
      - **Local Area Network (LAN)**
      - **Wide Area Network (WAN)**
      - **Metropolitan Area Network (MAN)**
      - **Personal Area Network (PAN)**
  - **Network Operating System** provides features between systems across network
    - **Communication scheme allows systems to exchange messages**
    - **Illusion of a single system**

Computing Environments – Client-Server

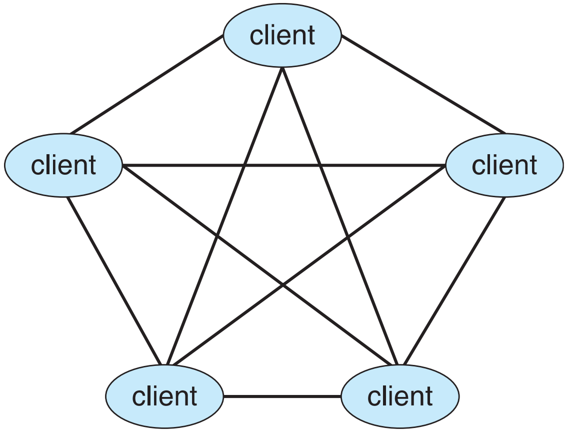
- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system provides an interface to client to request services (i.e., database)**
    - **File-server system provides interface for clients to store and retrieve files**



Computing Environments - Peer-to-Peer

- Another model of distributed system

- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - **Registers its service with central lookup service on network, or**
    - **Broadcast request for service and respond to requests for service via *discovery protocol***
  - Examples include Napster and Gnutella, **Voice over IP (VoIP)** such as Skype



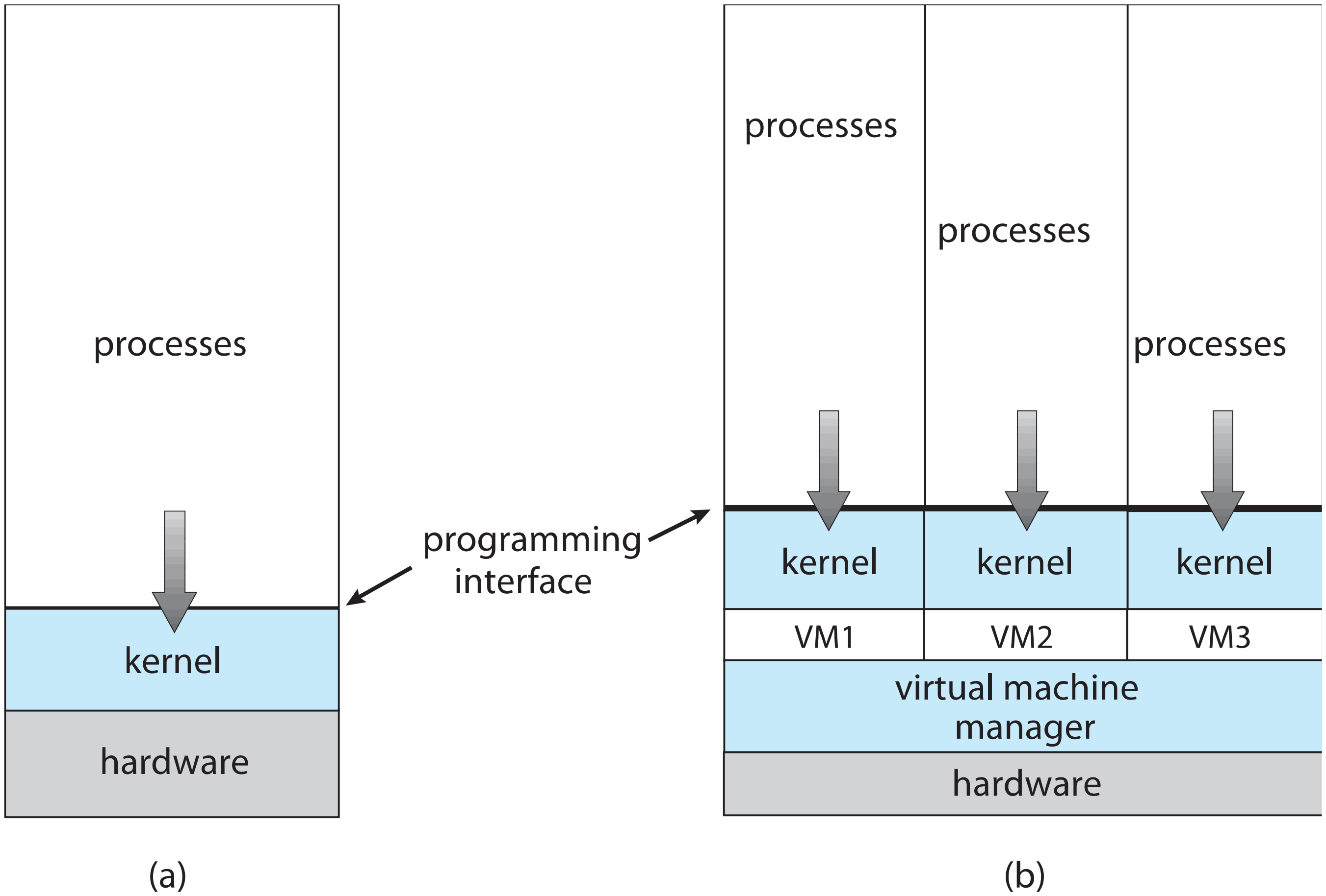
Computing Environments - Virtualization

- Allows operating systems to run applications within other OSes
  - Vast and growing industry
- **Emulation** used when source CPU type different from target type (i.e. PowerPC to Intel x86)
  - Generally slowest method
  - When computer language not compiled to native code – **Interpretation**
- **Virtualization** – OS natively compiled for CPU, running **guest** OSes also natively compiled
  - Consider VMware running WinXP guests, each running applications, all on native WinXP **host** OS
  - **VMM** (virtual machine Manager) provides virtualization services

Computing Environments - Virtualization

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
  - Apple laptop running Mac OS X host, Windows as a guest
  - Developing apps for multiple OSes without having multiple systems
  - QA testing applications without having multiple systems
  - Executing and managing compute environments within data centers
- VMM can run natively, in which case they are also the host
  - There is no general-purpose host then (VMware ESX and Citrix XenServer)

Computing Environments - Virtualization



Computing Environments – Cloud Computing

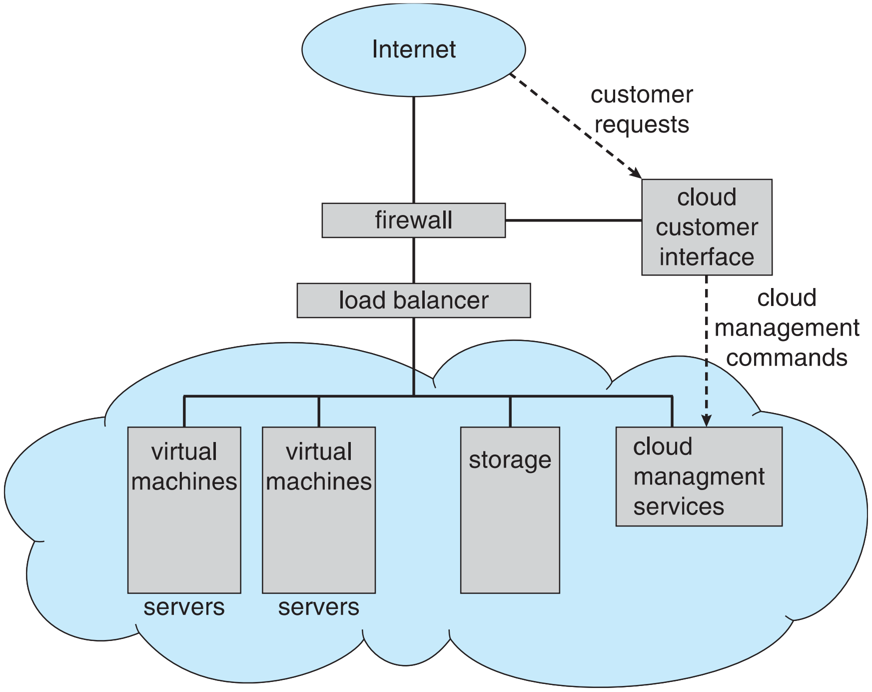
- Delivers computing, storage, even apps as a service across a network

- Logical extension of virtualization because it uses virtualization as the base for it functionality
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage

- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components
  - Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor)
  - Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server)
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use)

Computing Environments – Cloud Computing

- Cloud computing environments composed of traditional OSes, plus VMMs, plus cloud management tools
  - Internet connectivity requires security like firewalls
  - Load balancers spread traffic across multiple applications



Computing Environments – Real-Time Embedded Systems

- Real-time embedded systems most prevalent form of computers
  - Vary considerable, special purpose, limited purpose OS, **real-time OS**
  - Use expanding
- Many other special computing environments as well
  - Some have OSes, some perform tasks without an OS
- Real-time OS has well-defined fixed time constraints
  - Processing ***must*** be done within constraint
  - Correct operation only if constraints met

Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary **closed-source**
- Counter to the **copy protection** and **Digital Rights Management (DRM)** movement
- Started by **Free Software Foundation (FSF)**, which has "copyleft" **GNU Public License (GPL)**
- Examples include **GNU/Linux** and **BSD UNIX** (including core of **Mac OS X**), and many more
- Can use VMM like VMware Player (Free on Windows), Virtualbox (open source and free on many platforms - http://www.virtualbox.com)
  - Use to run guest operating systems for exploration