CS307&CS356: Operating Systems

Dept. of Computer Science & Engineering
Chentao Wu 吴晨涛
wuct@cs.sjtu.edu.cn

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Download lectures

- ftp://public.sjtu.edu.cn
- User: wuct
- Password: wuct123456

- http://www.cs.sjtu.edu.cn/~wuct/os/

Brief Introduction

- Work & Education Experience
  - 2014-present, associate professor, Dept. of Computer Science & Engineering, *Shanghai Jiao Tong University (SJTU)* , Shanghai, China
  - 2012-2014, assistant professor, Dept. of Computer Science & Engineering, *Shanghai Jiao Tong University (SJTU)* , Shanghai, China
  - 2012, Ph.D., Electrical and Computer Engineering, *Virginia Commonwealth University (VCU)*, Richmond, VA, USA
  - 2010, Ph.D., Computer Architecture, *Huazhong University of Science and Technology (HUST)*, Wuhan, China

- Research Interest: Big Data/Cloud Storage Systems

Welcome to Join in Our Lab

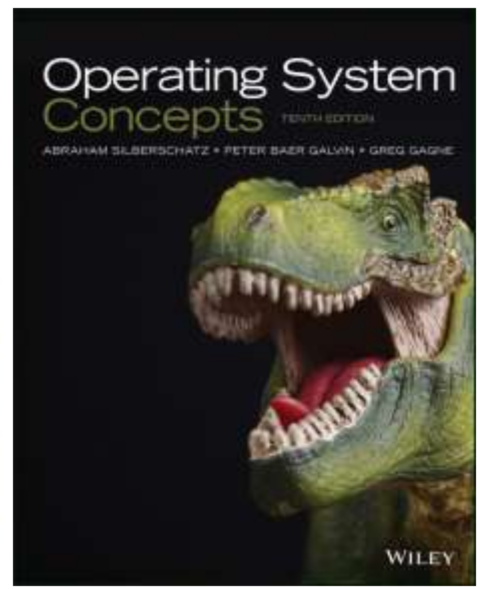
- Look for candidates under my supervision: 2-3 master/1-2 Ph.D. students per year
- From 2nd year to 4th year
- Research on Big Data/Cloud Storage Systems
  - **Chapters 9-15 in OS book**
  - Cloud Storage/Big Data storage devices (NAS/SAN/RAID)
  - Data Management (e.g., cache, I/O scheduling)
  - Non-Volatile Memories (e.g. Flash, Phase Change Memory, etc.)
  - Distributed File Systems (e.g., HDFS, Ceph)

OS Textbooks (Old)

- Operating System Concepts (7^{th} Edition)
  - A. Silberschatz
  - P. Galvin
  - G. Gagne
  - ISBN: 978-7-040-20928-0

OS Textbooks (New)

- Operating System Concepts (9th & 10th Edition)
  - A. Silberschatz
  - P. Galvin
  - G. Gagne
  - ISBN: 978-7-111-60436-5



Textbooks (Electronic)

- Electronic Files of Books are available in the FTP
  - 8th Edition of the Operating System Book
  - 9th Edition of the Operating System Book
  - 10th Edition of the Operating System Book
- Due to the copyright policies, please DO NOT spread the PDF files.

Syllabus (1)

- Requirements:
  - Computer Basic
  - C/C++ Programming
- Goals: Successful course participants will:
  - Understand basic machine organization, including processors, main memory, and input/output architecture.
  - Understand the basics of the memory hierarchy, including virtual memory and caches, and how these are implemented in hardware and software.

Syllabus (2)

- Goals (contd.)
  - Understand the core concepts of operating systems, including processes, threads, synchronization, virtual memory policies, and file management.
  - The idea of the course is to learn how computers really work, from the chip level up to the application level. When we finish, you will understand what is actually happening when a computer system is running a set of programs, and will be able to make informed choices as a developer, project manager, or system customer.

Course Meeting Time

- Lectures:
  - 4 classes per week
- Questions:
  - Ask me directly between/after the classes
  - Go to my office: SEIEE 3-513
  - Send me an email: wuct@cs.sjtu.edu.cn

Final Grades

- Homework and Attendance 10%
  - Weekly
- Quizzes 15% (Close Book, On Classes)
  - Will be announced before two weeks
  - Three quizzes
- Project 15% (Presentation and Report)
  - Some students will be selected to give presentation on classes
- Final Exam 60% (Close Book)

Quizzes

- 1. Schedule
  - **First Quiz: Mar. 28th (Thursday, 5th Week)**
  - Chapter 1-3 in OS book
  - **Second Quiz: Apr. 18th (Thursday, 8th Week)**
  - Chapter 4-7 in OS book
  - **Third Quiz: May 9th (Thursday, 11th Week)**
  - Chapter 8-10 in OS book
- 2. Scores
  - Each quiz has 10 points, half points of each quiz will be calculated as a part of the final grade.

Late Policy

- Late Policy: Deadlines will be given in each assignment. These deadlines are strict.
- Typically, homework will be given on each Monday, you should submit your homework by the next Monday.

Projects

- 4 categories of projects (15% in the Final Grade):
  - Project 1: 3%
  - Project 2-3: 4%
  - Project 4-6: 4%
  - Project 7-8: 4%
- Arrangement for computer room: TBA

Project 1

- **Project 1: Introduction to Linux Kernel Modules**
- At the end of Chapter 2 (P1-P7, textbook)
  - Deadline: Oct. 19th (Friday in the 6th Week)

Project 2

- **Project 2-1: UNIX Shell**
- **Project 2-2: Linux Kernel Module for Task Information**
- At the end of Chapter 3(P12-P22, textbook)
  - Deadline: Nov. 2nd (Friday in the 8th Week)

Project 3

- **Project 3-1: Multithreaded Sorting Application**
- **Project 3-2: Fork-Join Sorting Application**
- At the end of Chapter 4(P25-P28, textbook)
  - Deadline: Nov. 2nd (Friday in the 8^{th} Week)

Project 4

- **Project 4: Scheduling Algorithms**
- At the end of Chapter 5(P29-P31, textbook)
  - Deadline: Nov. 16th (Friday in the 10th Week)

Project 5

- **Project 5-1: Designing a Thread Pool**
- **Project 5-2: The Producer – Consumer Problem**
- At the end of Chapter 7(P35-P44, textbook)
  - Deadline: Nov. 16th (Friday in the 10^{th} Week)

Project 6

- **Project 6: Banker's Algorithm**
- At the end of Chapter 8(P45-P47, textbook)
  - Deadline: Nov. 16th (Friday in the 10^{th} Week)

Project 7

- **Project 7: Contiguous Memory Allocation**
- At the end of Chapter 9(P48-P50, textbook)
  - Deadline: Nov. 30th (Friday in the 12^{th} Week)

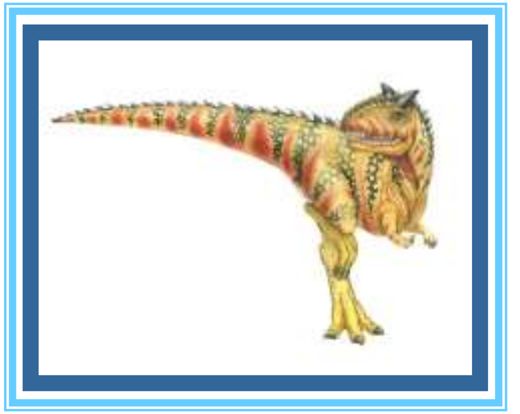
Project 8

- **Project 8: Designing a Virtual Memory Manager**
- At the end of Chapter 10(P51-P54, textbook)
  - Deadline: Nov. 30th (Friday in the 12th Week)

Teaching Assistants

- Xin Xie(谢鑫)
  - Email: 1033556059@qq.com
  - Mobile Phone: 15221833021
- Huayi Jin(金华溢)
  - Email: 1298222739@qq.com
  - Mobile Phone: 13701857162

Chapter 1: Introduction



Chapter 1: Introduction

- What Operating Systems Do
- Computer-System Organization
- Computer-System Architecture
- Operating-System Operations
- Resource Management
- Security and Protection
- Virtualization
- Distributed Systems
- Kernel Data Structures
- Computing Environments
- Free/Libre and Open-Source Operating Systems

Objectives

- Describe the general organization of a computer system and the role of interrupts
- Describe the components in a modern, multiprocessor computer system
- Illustrate the transition from user mode to kernel mode
- Discuss how operating systems are used in various computing environments
- Provide examples of free and open-source operating systems

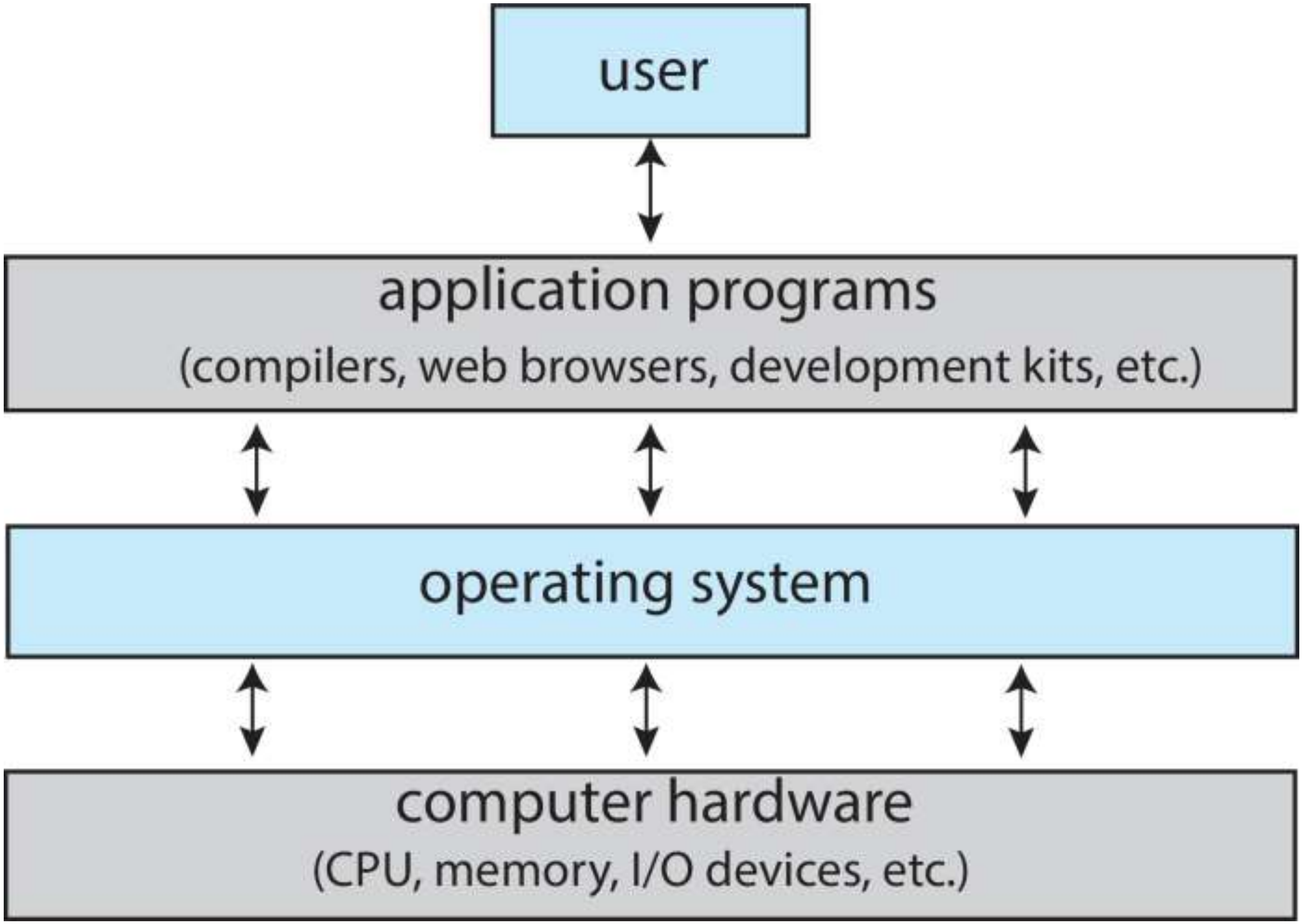
Computer System Structure

- 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

4层

  - 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

Abstract View of Components of Computer



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
  - 分配
  - Operating system is a **resource allocator** and **control program** making efficient use of HW and managing execution of user programs
- Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**
- Mobile devices like smartphones and tables are resource poor, optimized for usability and battery life
  - Mobile user interfaces such as touch screens, voice recognition
- Some computers have little or no user interface, such as embedded computers in devices and automobiles
  - Run primarily without user intervention

Defining Operating Systems

- Term OS covers many roles
  - Because of myriad designs and uses of OSes
  - Present in toasters through ships, spacecraft, game machines, TVs and industrial control systems 控制程序
  - Born when fixed use computers for military became more general purpose and needed resource management and program control 资源分配器

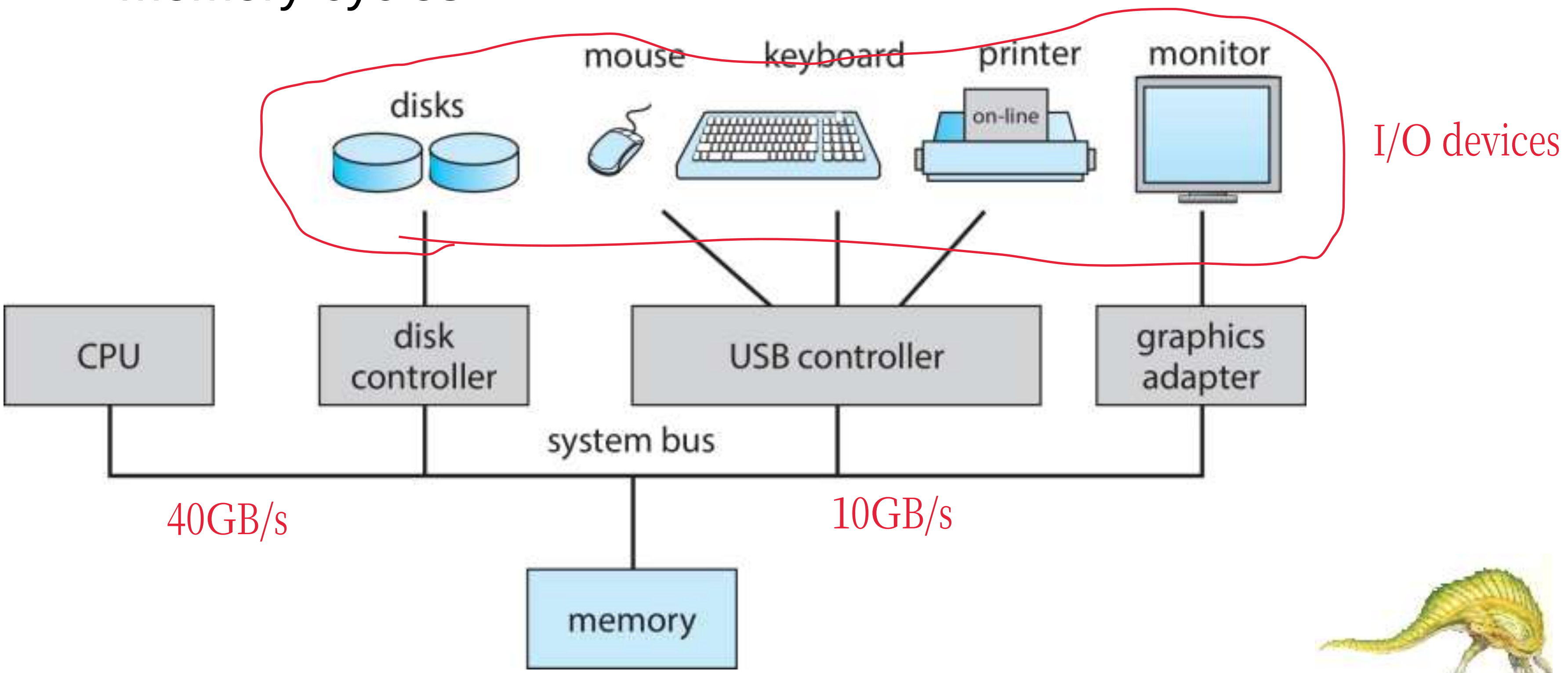
Operating System Definition (Cont.)

- 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,** part of the operating system
- Everything else is either
  - a **system program** (ships with the operating system, but not part of the kernel) , or

    系统程序不在操作系统内核中，不是核心功能（如图形化界面）
  - an **application program**, all programs not associated with the operating system
- Today’s OSes for general purpose and mobile computing also include **middleware** – a set of software frameworks that provide addition services to application developers such as databases, multimedia, graphics

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 每一种硬件设备如硬盘上都有自己的CPU



Computer-System Operation

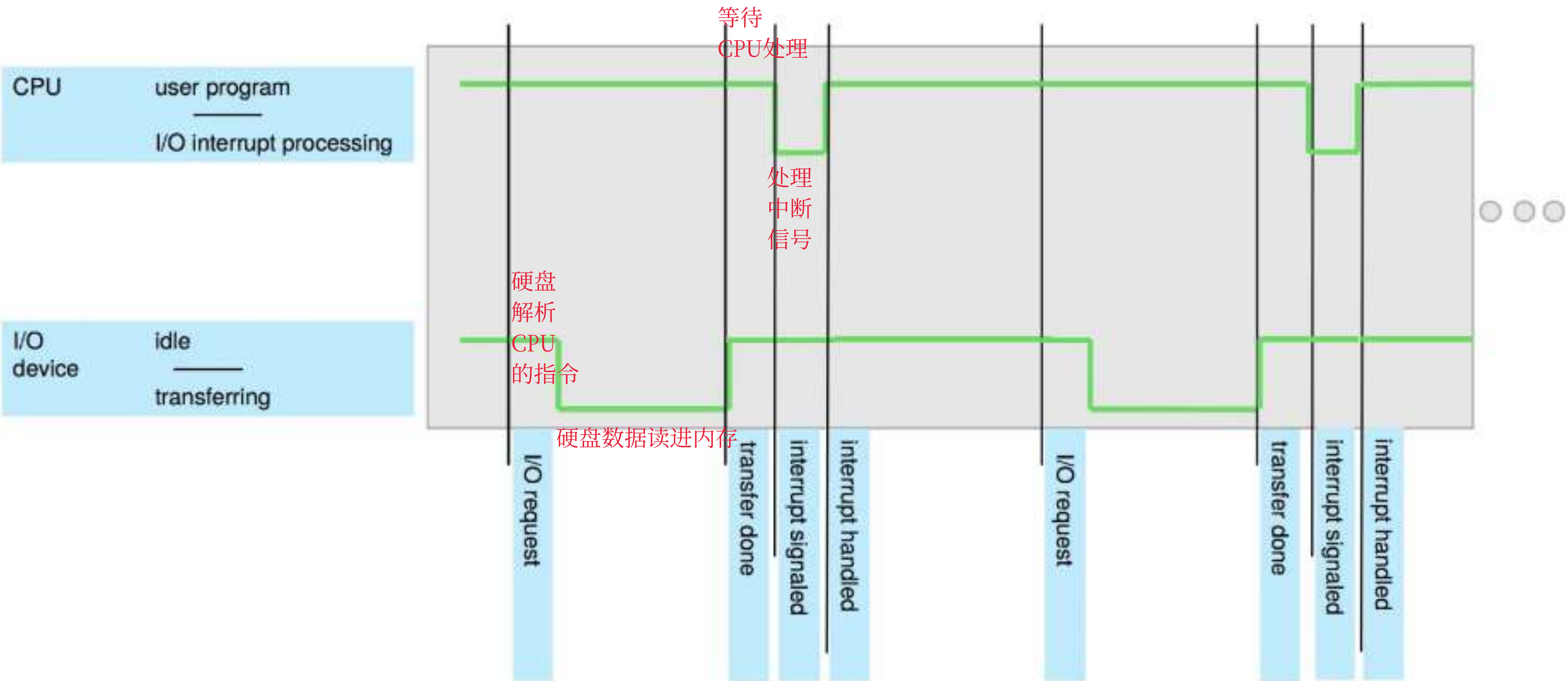
- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- Each device controller type has an operating system **device driver** to manage it
- CPU moves data from/to main memory to/from local buffers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

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 中断向量表
- 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 软件中断
- An operating system is **interrupt driven**

Interrupt Timeline

硬件中断：任务完成
软件中断：数据溢出、异常



Computer Startup
启动

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware** 固件
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

刷BIOS升级：跳电

BIOS写在ROM上，固定在主板上

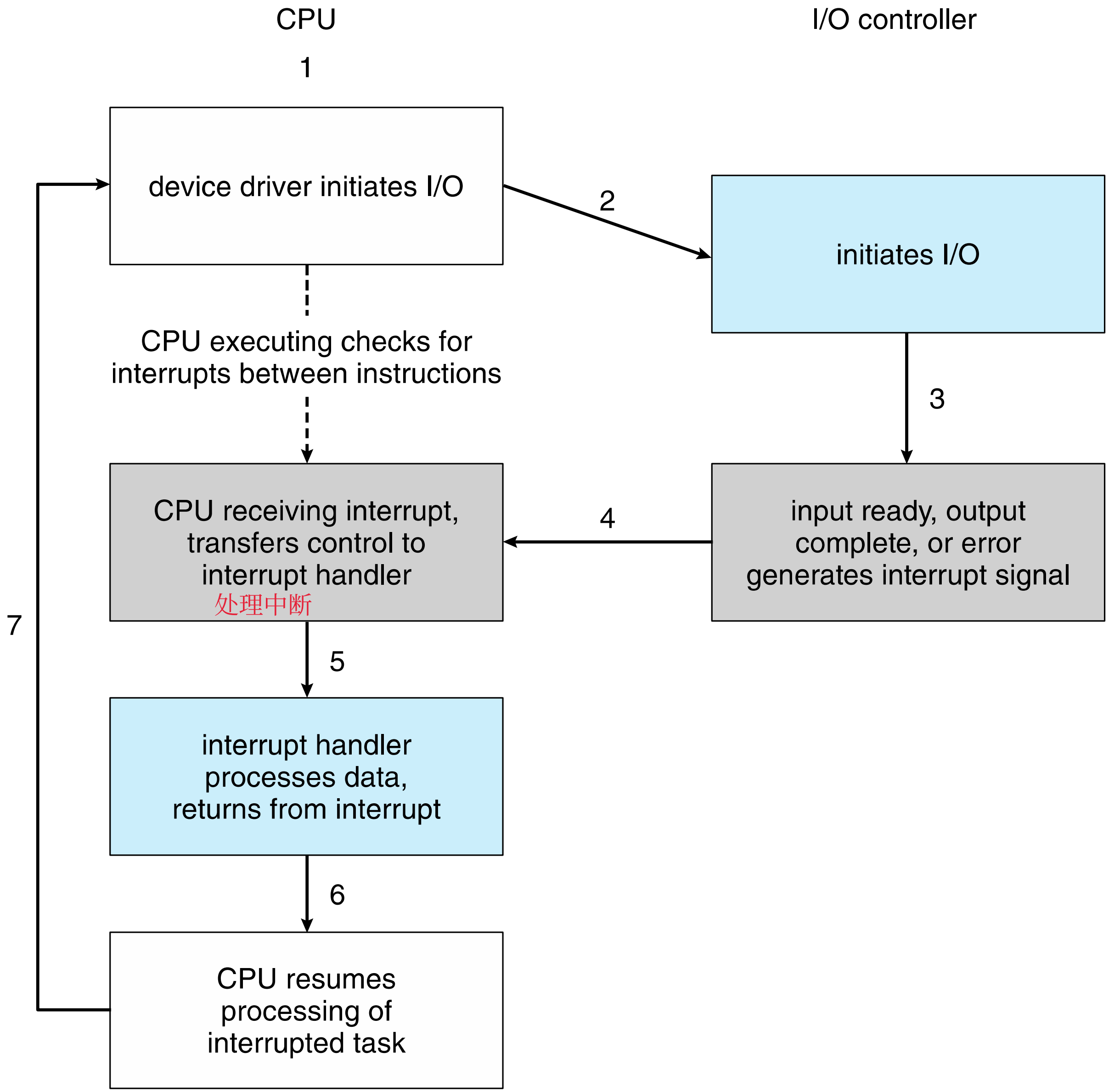
ROM: read only memory只读存储器
RAM: random access memory随机访问存储器

1.F1 ->BIOS (basic input/output system)基本输入输出系统
从哪个盘启动 CD/DVD-ROM/USB/HDD/SSD
2.GRUB 系统启动选项条 选择进入哪个系统
3.HDD/SSD ：Master Boot Recorder 引导扇区 找到系统安装的位置分区
4.Initialize OS 初始化系统

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-drive I/O Cycle



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

上层自己写编程如C
语言的 read（）
->OS — read
device
driver

- After I/O starts, control returns to user program without waiting for I/O completion
  - 系统调用
    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 Structure

- Main memory – only large storage media that the CPU can access directly
  - **Random access**
  - Typically **volatile**
  - Typically **random-access memory** in the form of **Dynamic Random-access Memory (DRAM)**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- **Hard Disk Drives** (**HDD**) – 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
- **Non-volatile memory (NVM)** devices– faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular as capacity and performance increases, price drops

Storage Definitions and Notation Review

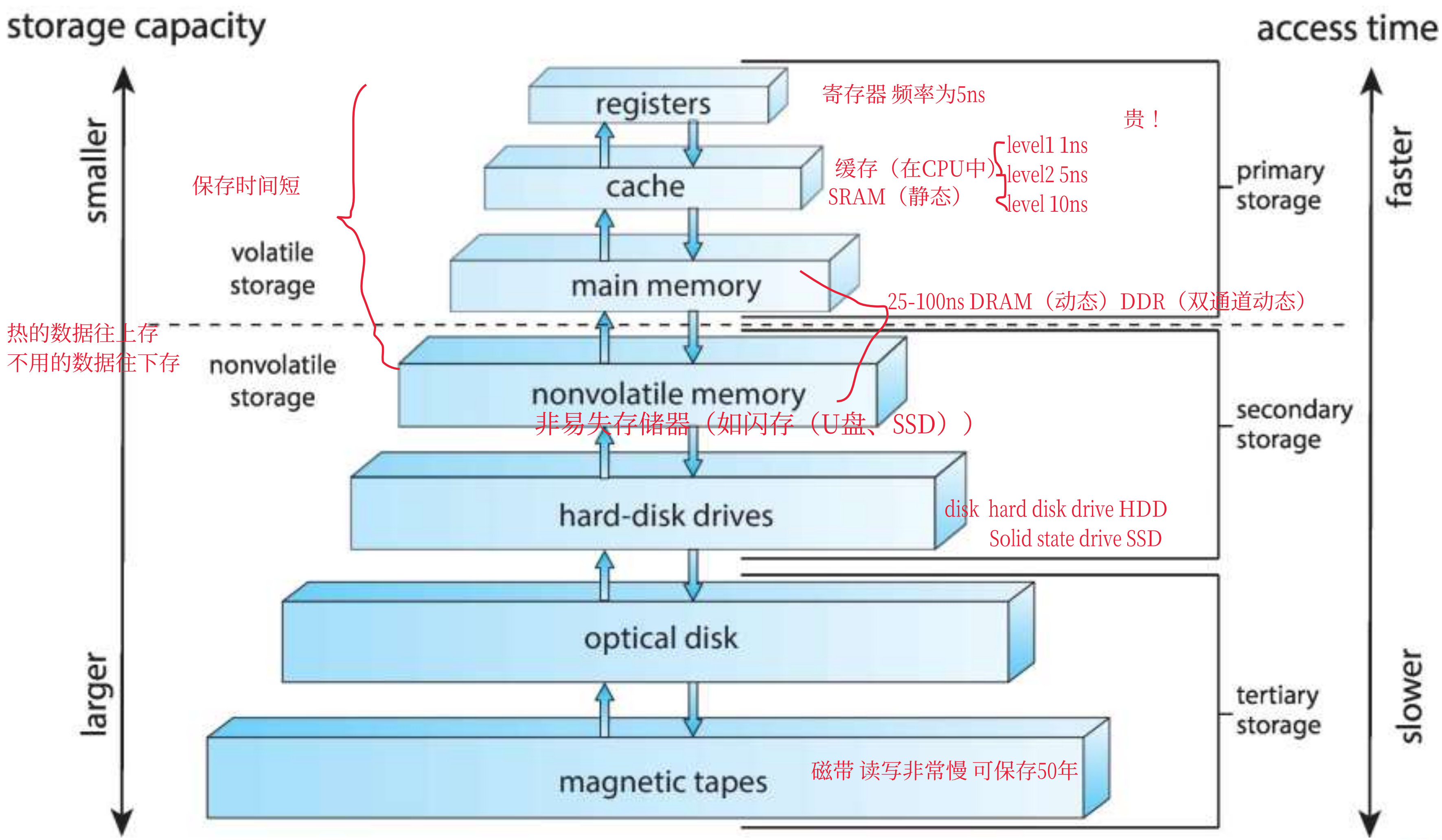
The basic unit of computer storage is the **bit** . A bit can contain one of two values, 0 and 1. All other storage in a computer is based on collections of bits. Given enough bits, it is amazing how many things a computer can represent: numbers, letters, images, movies, sounds, documents, and programs, to name a few. A **byte** is 8 bits, and on most computers it is the smallest convenient chunk of storage. For example, most computers don't have an instruction to move a bit but do have one to move a byte. A less common term is **word**, which is a given computer architecture's native unit of data. A word is made up of one or more bytes. For example, a computer that has 64-bit registers and 64-bit memory addressing typically has 64-bit (8-byte) words. A computer executes many operations in its native word size rather than a byte at a time.

Computer storage, along with most computer throughput, is generally measured and manipulated in bytes and collections of bytes. A **kilobyte** , or KB , is 1,024 bytes; a **megabyte** , or **MB** , is $1{,}024^2$ bytes; a **gigabyte** , or GB , is $1{,}024^3$ bytes; a **terabyte** , or **TB** , is $1{,}024^4$ bytes; and a **petabyte** , or **PB** , is $1{,}024^5$ bytes. Computer manufacturers often round off these numbers and say that a megabyte is 1 million bytes and a gigabyte is 1 billion bytes. Networking measurements are an exception to this general rule; they are given in bits (because networks move data a bit at a time).

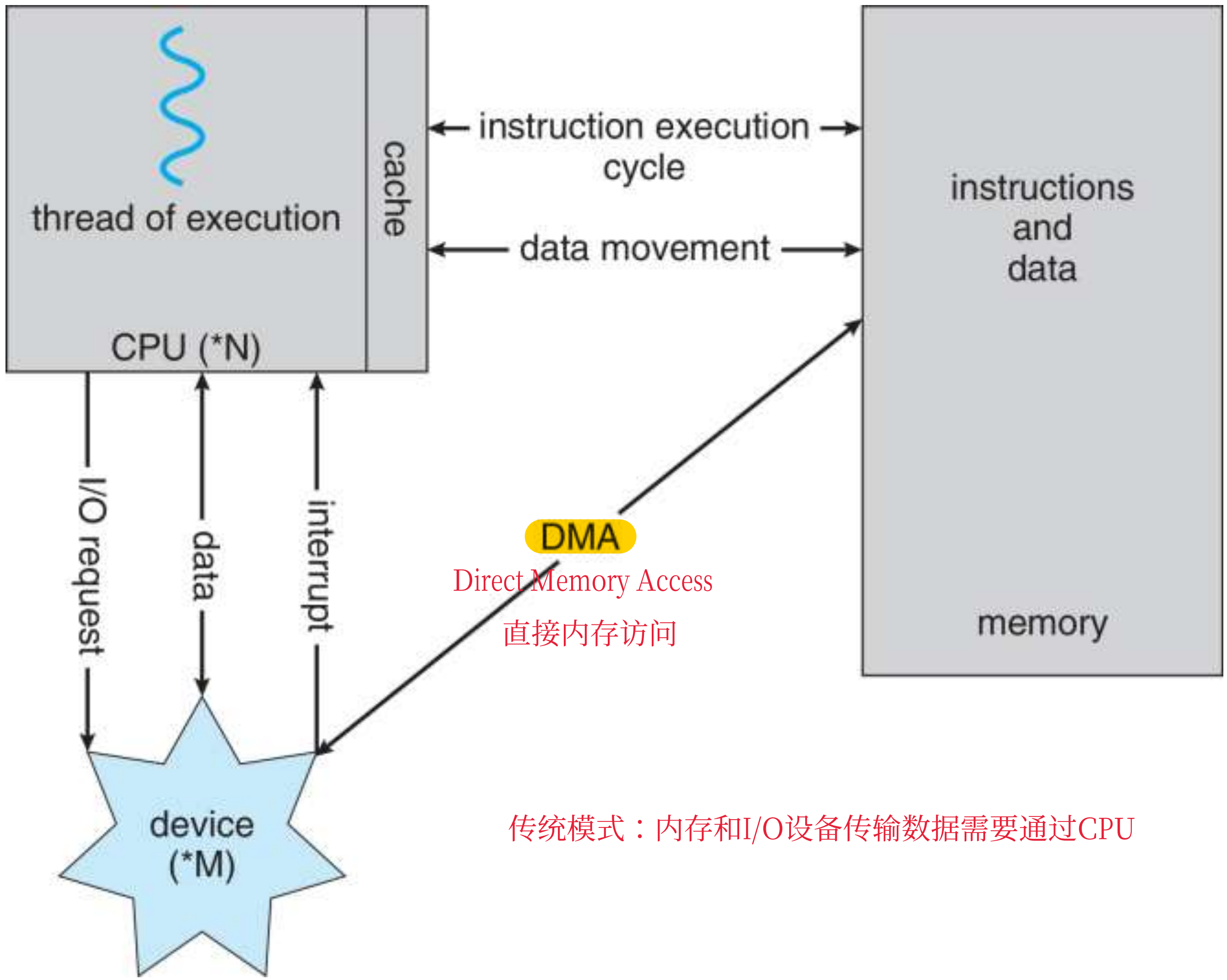
Storage Hierarchy

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



How a Modern Computer Works



传统模式：内存和I/O设备传输数据需要通过CPU

A von Neumann architecture

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 缓存 storage directly to main memory without CPU intervention 主存
- Only one interrupt is generated per block, rather than the one interrupt per byte

缺点：可靠性（稳定性）
传统模式中CPU可判断数据的正确性，如果出错可以重新传输

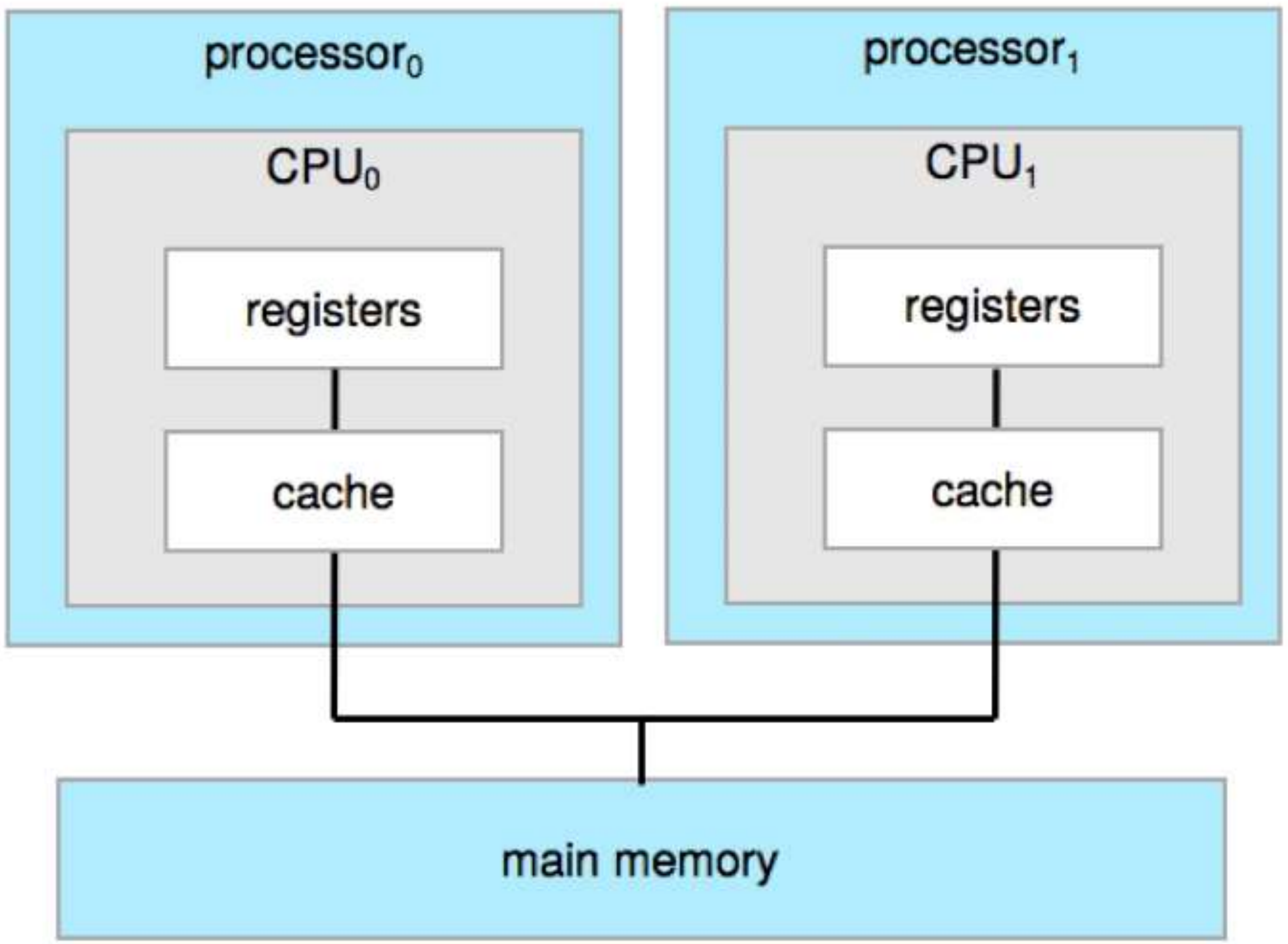
Computer-System Architecture

- 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**
    并行系统
  - Advantages include:
    1. **Increased throughput** 吞吐量
    2. **Economy of scale**
    3. **Increased reliability** – graceful degradation or fault tolerance
  - Two types:
    1. 非对称 **Asymmetric Multiprocessing** – each processor is assigned a specie task.
    2. **Symmetric Multiprocessing** – each processor performs all tasks
       对称
       所有处理器型号一样，能力一样

multiprocessor:多处理器
multi-core:多核（CPU内部而言）（十个以下）
many-core：众核（十个以上）
一个处理器里可能有几个核

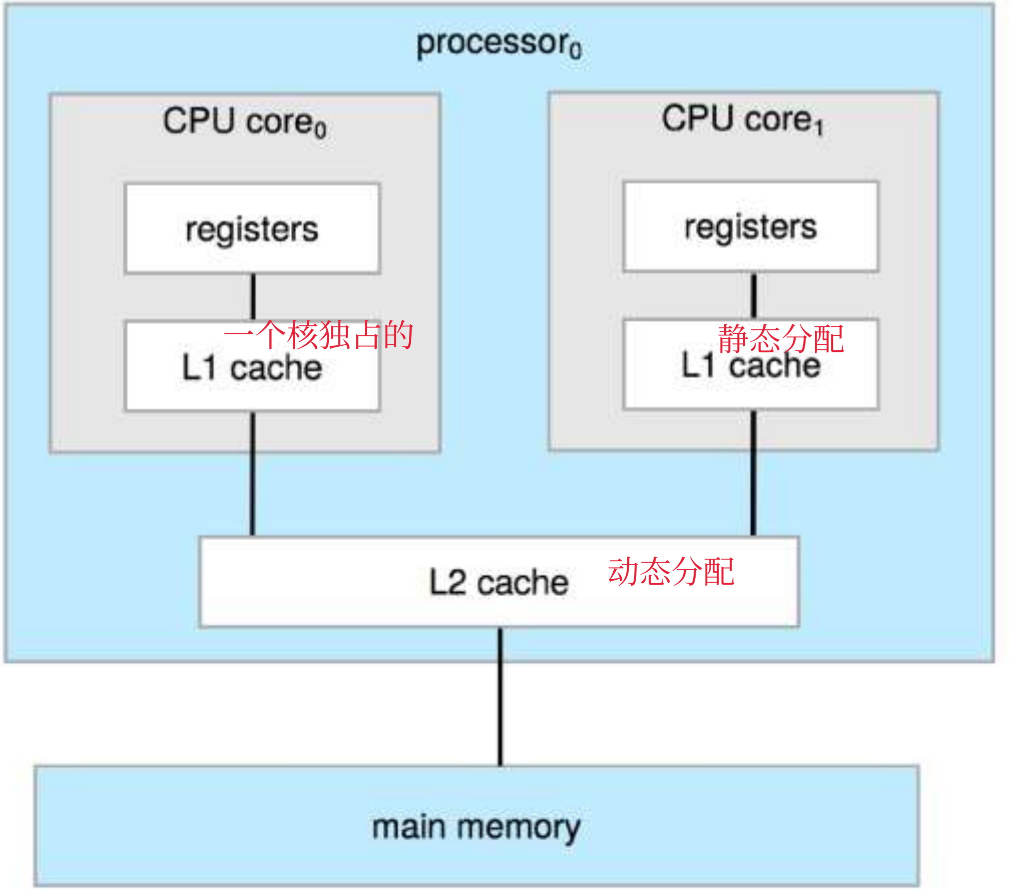
distributed systems:分布式系统
一台机器中足够多的CPU或内存使总线带宽不够，于是用多台机器或服务器，之间用网络连接

Symmetric Multiprocessing Architecture



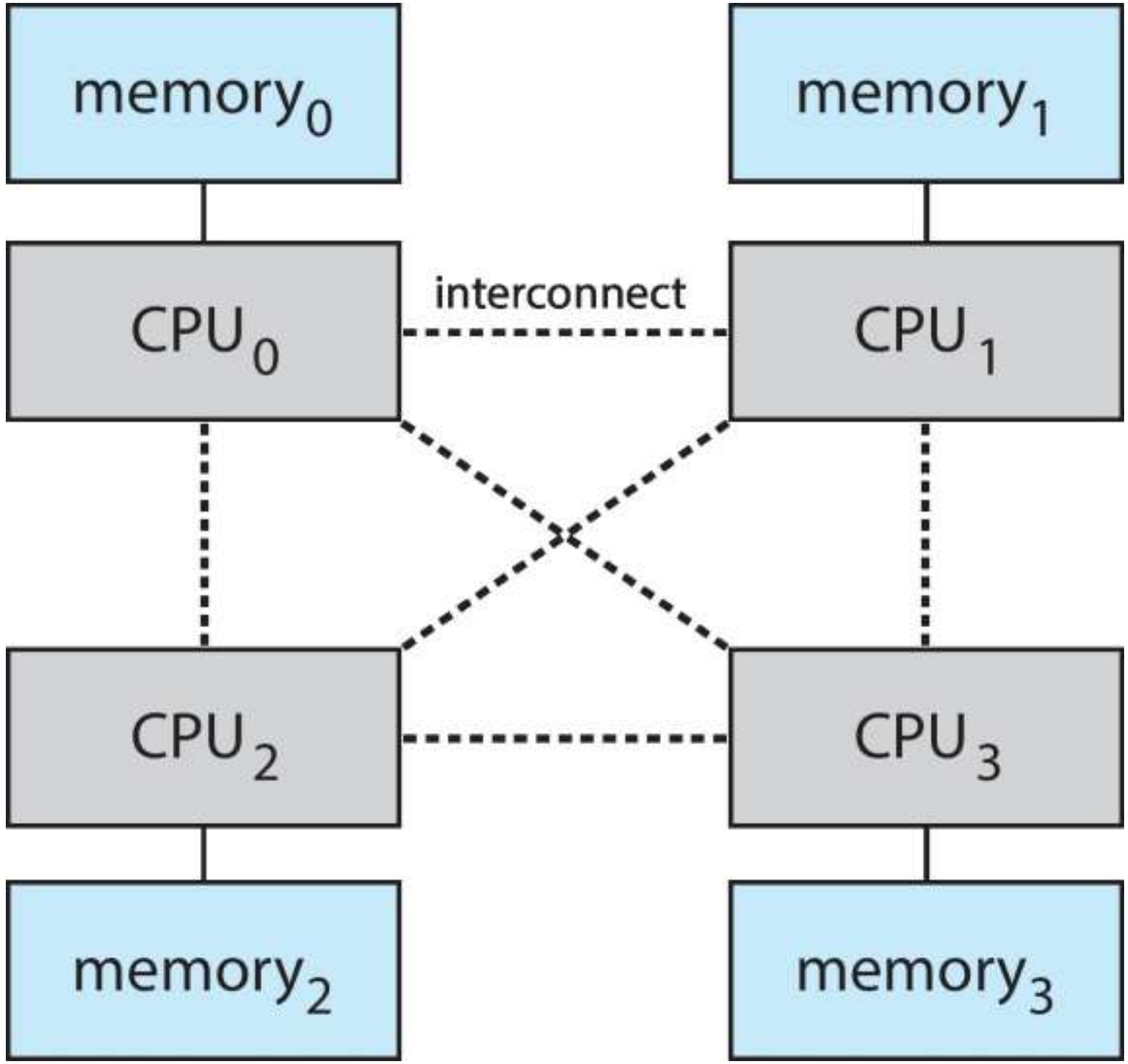
A Dual-Core Design

- Multi-chip and **multicore**
- Systems containing all chips
  - Chassis containing multiple separate systems



非对称式内存访问（NUMA）

Non-Uniform Memory Access System



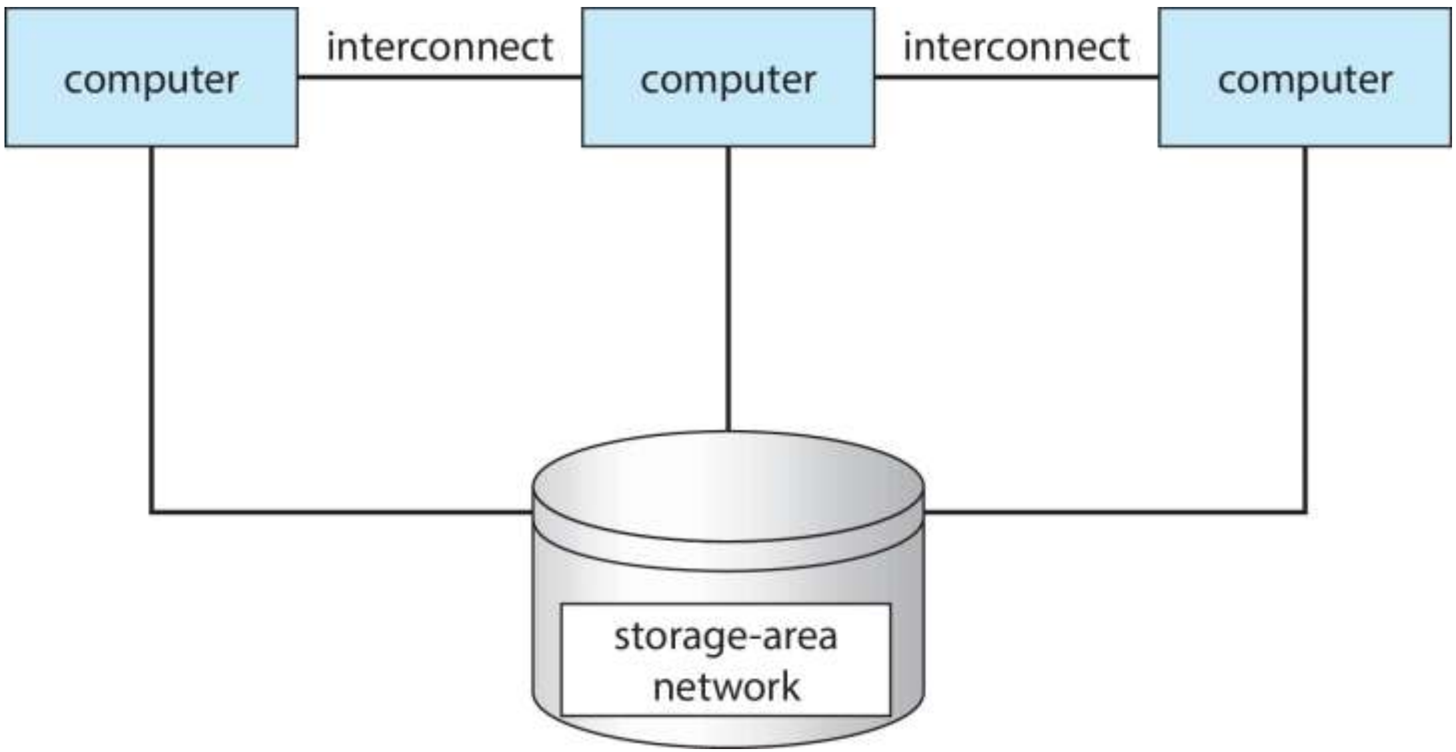
每个CPU访问自己的内存是快速访问
访问其他CPU的内存是慢速访问

Clustered Systems

集群（100台服务器以内）

- 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)** 高性能计算
    - Applications must be written to use **parallelization**
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

Clustered Systems

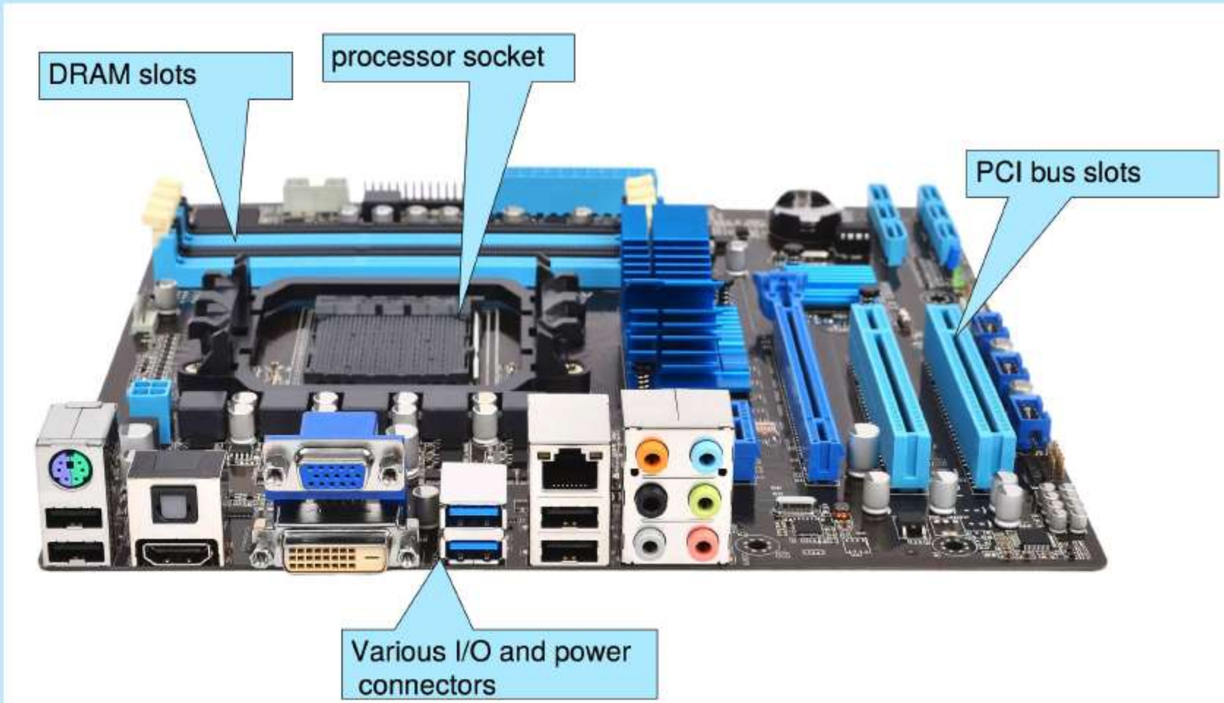


存储局域网

主板

PC Motherboard

Consider the desktop PC motherboard with a processor socket shown below:



This board is a fully-functioning computer, once its slots are populated. It consists of a processor socket containing a CPU, DRAM sockets, PCIe bus slots, and I/O connectors of various types. Even the lowest-cost general-purpose CPU contains multiple cores. Some motherboards contain multiple processor sockets. More advanced computers allow more than one system board, creating NUMA systems.

Operating-System Operations

- Bootstrap program – simple code to initialize the system, load the kernel
- Kernel loads
- Starts **system daemons** (services provided outside of the kernel)
- Kernel **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 – **system call**
    - Other process problems include infinite loop, processes modifying each other or the operating system

Multiprogramming and Multitasking

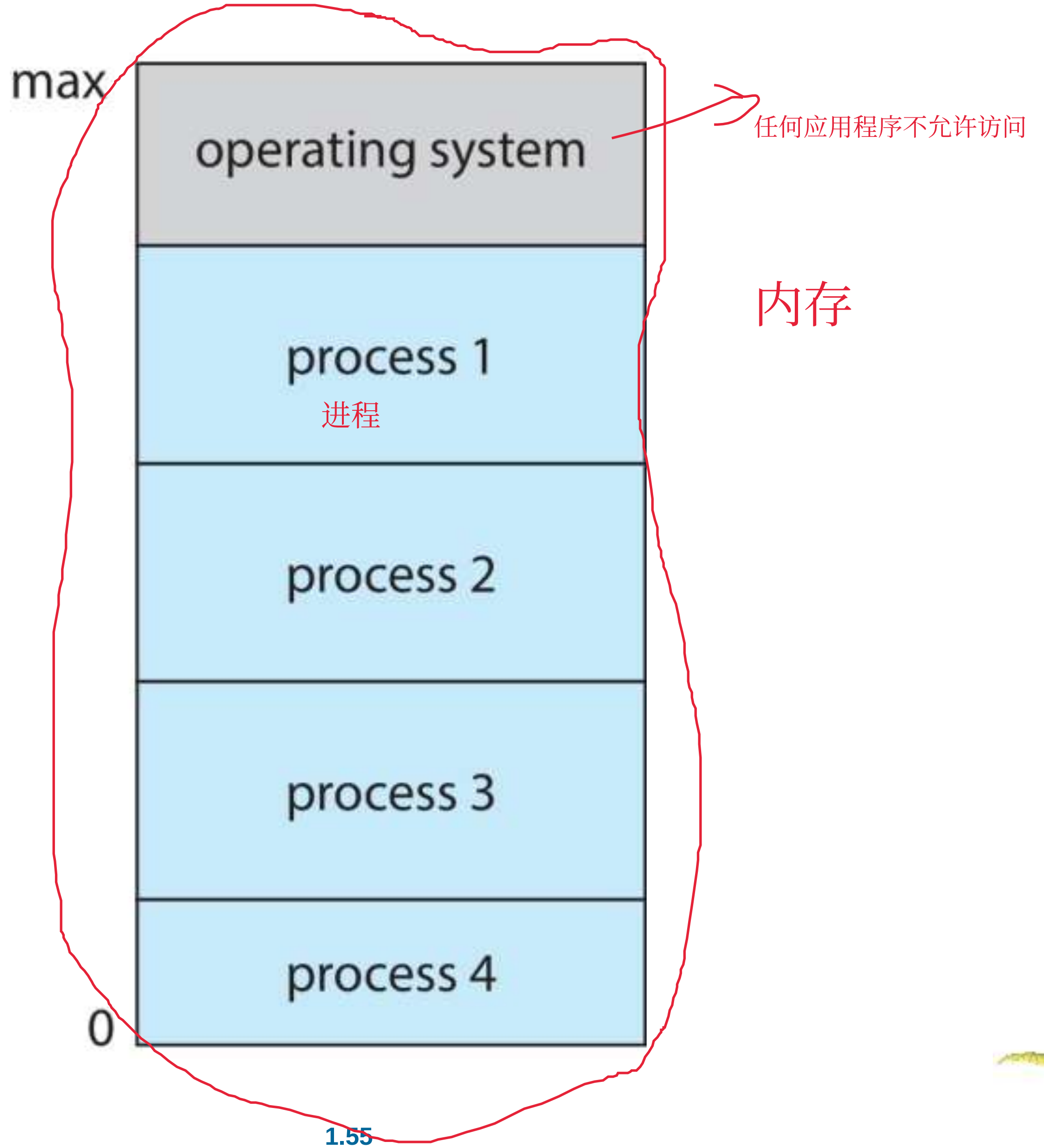
多任务并行 分批操作系统

- **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** 优先级
  - When it has to wait (for I/O for example), OS switches to another job

分时操作系统

- **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 把CPU分成时间片time slides/time quantum，各任务轮流执行如100ms
  - **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 数据交换
  - **Virtual memory** allows execution of processes not completely in memory

Memory Layout for Multiprogrammed System

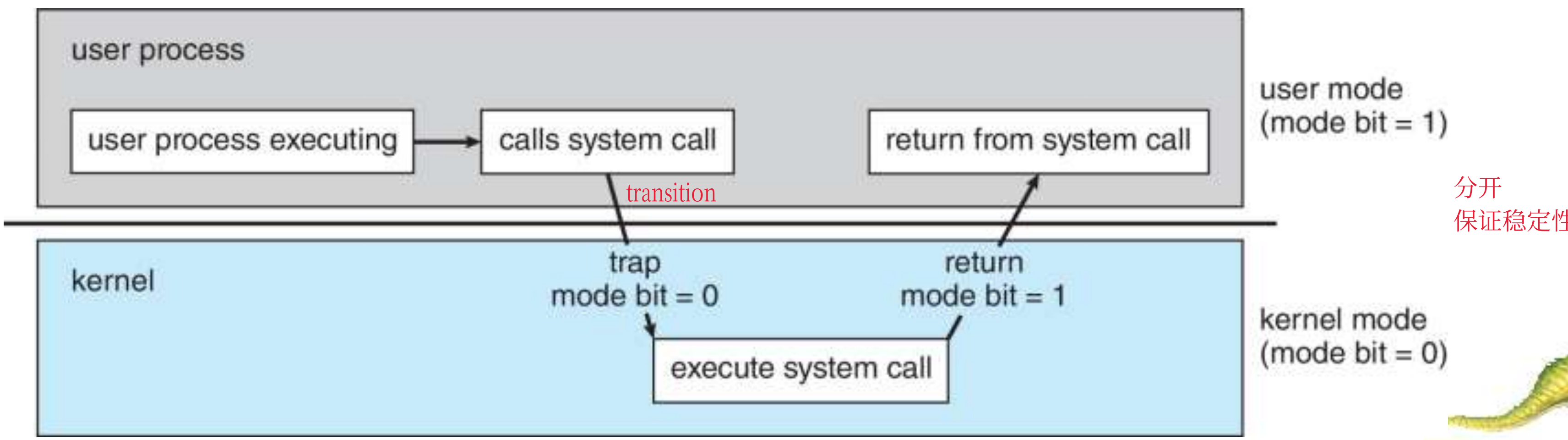


Dual-mode and Multimode Operation

- **Dual-mode** operation allows OS to protect itself and other system components
  - 用户态 内核态 保证应用程序和操作系统的稳定性
  - **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
死锁

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

File-system 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
  - Mounting and unmounting
  - Free-space management
  - Storage allocation
  - Disk scheduling
  - Partitioning
  - Protection
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications

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

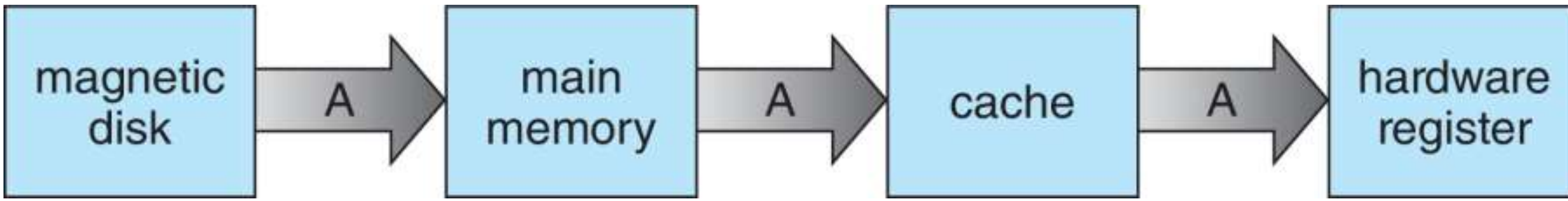
Characteristics of Various Types of Storage

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 read:25微秒 write：100微秒	magnetic disk
Access time (ns)	0.25-0.5	0.5-~~25~~ 0.5-10	~~80-250~~ 25-100	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 层级



- 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 19

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

不讲

- **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

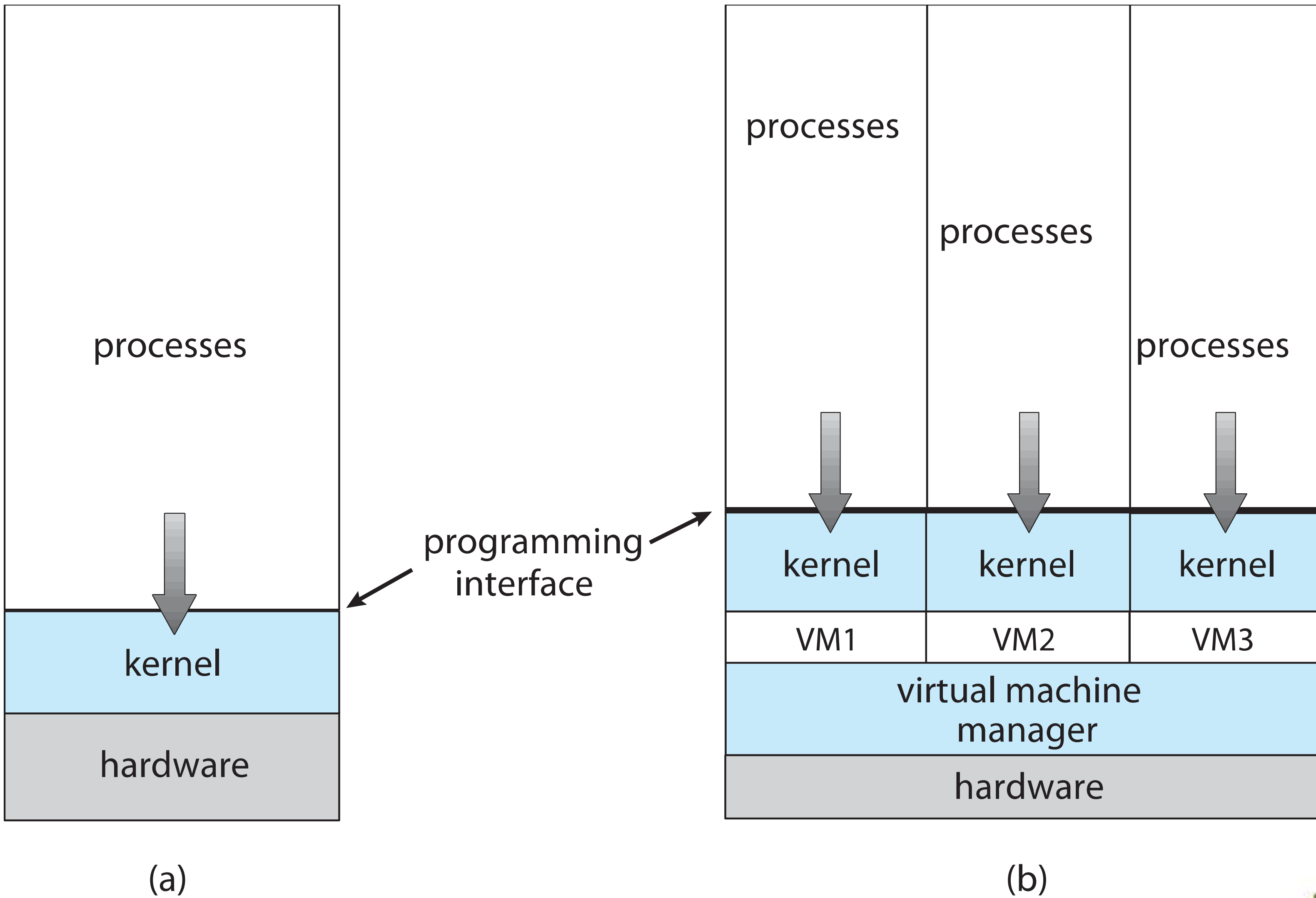
Virtualization

- 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

Virtualization (cont.)

- 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

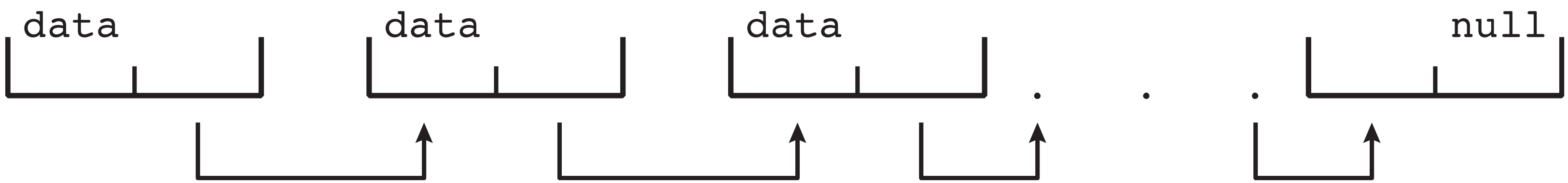


Distributed Systems

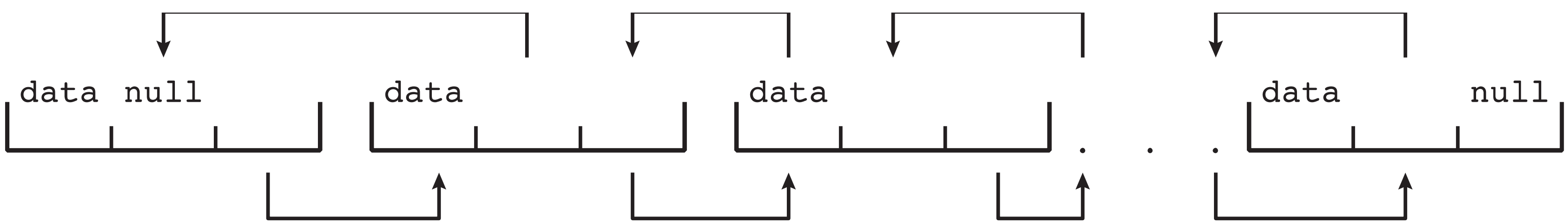
- Distributed computiing
  - Collection of separate, possibly heterogeneous, systems networked together
    - **Network** is a communications path, **TCP/IP** most common
      - **Local Area Network** (**LAN**) 局域网
      - **Wide Area Network** (**WAN**) 广域网
      - **Metropolitan Area Network** (**MAN**)
      - **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

Kernel Data Structures

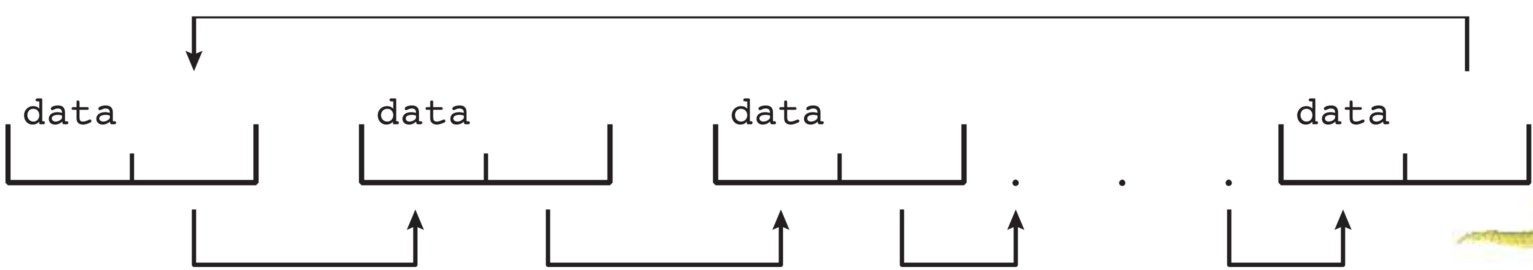
- Many similar to standard programming data structures
- ***Singly linked list***



- ***Doubly linked list***

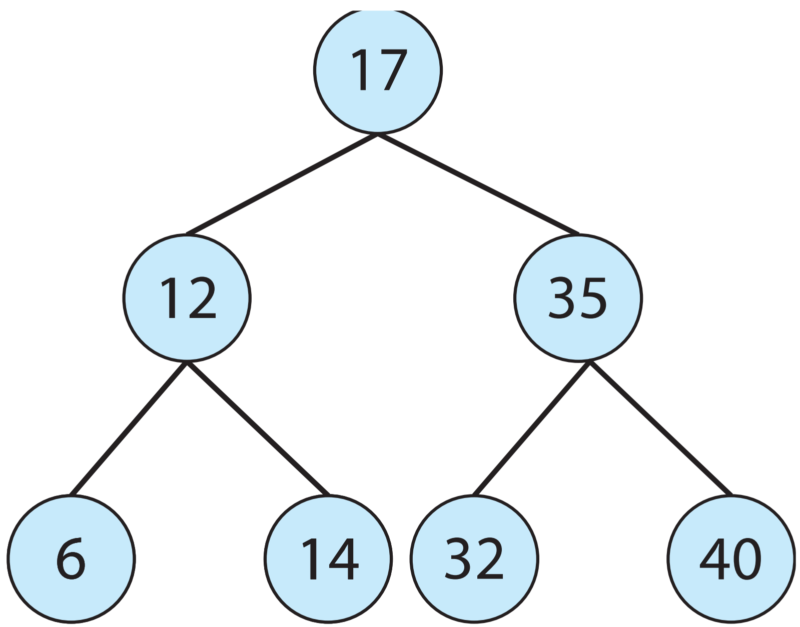


- ***Circular linked list***



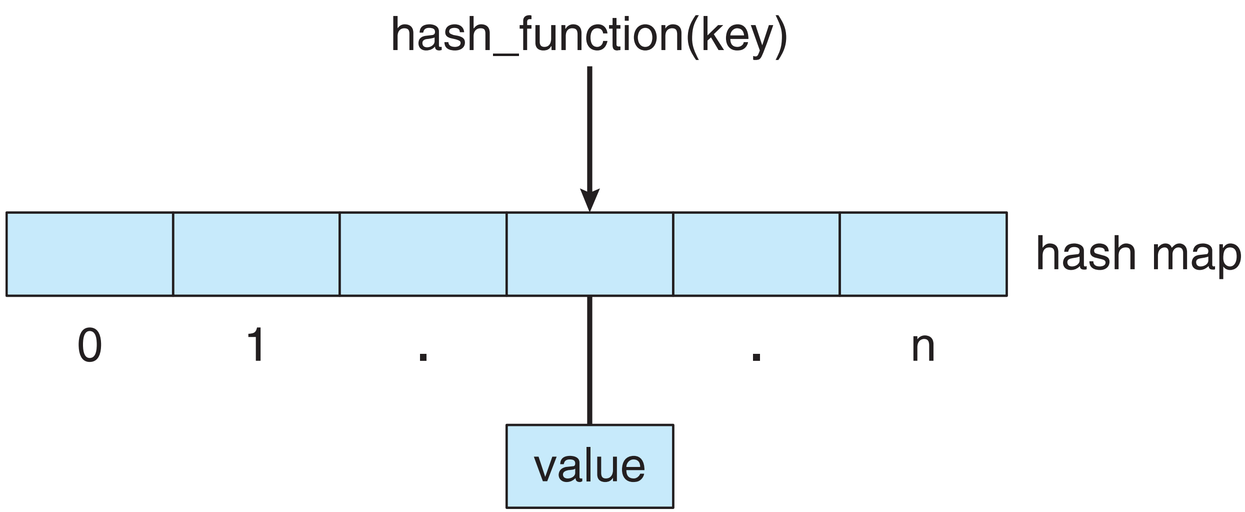
Kernel Data Structures

- **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

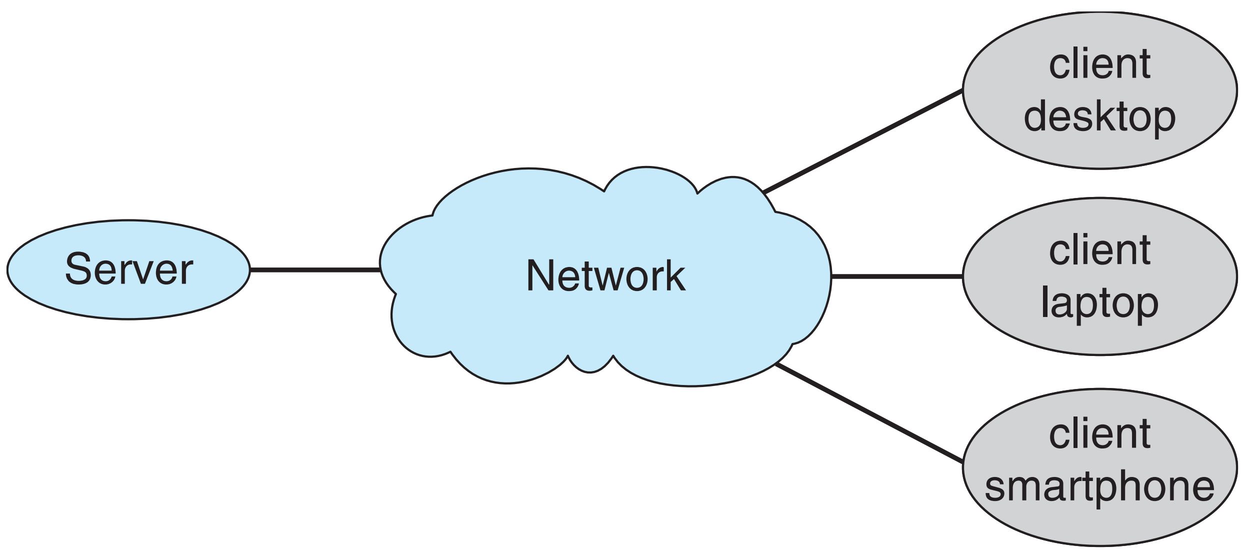
Computing Environments - Mobile

- 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 – Client-Server

客户端服务器模式（Client-Server)

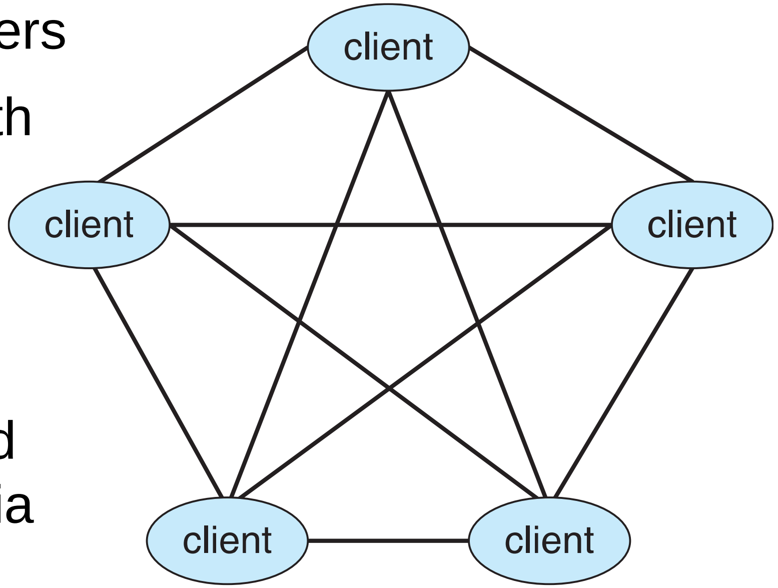
- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services (i.e., database)
    - **File-server system** provides interface for 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***协议
  - Examples include Napster and Gnutella, **Voice over IP** (**VoIP**) such as Skype
    基于IP的语音传输

即是client，又充当server
下载了的数据可以供给其他人下载

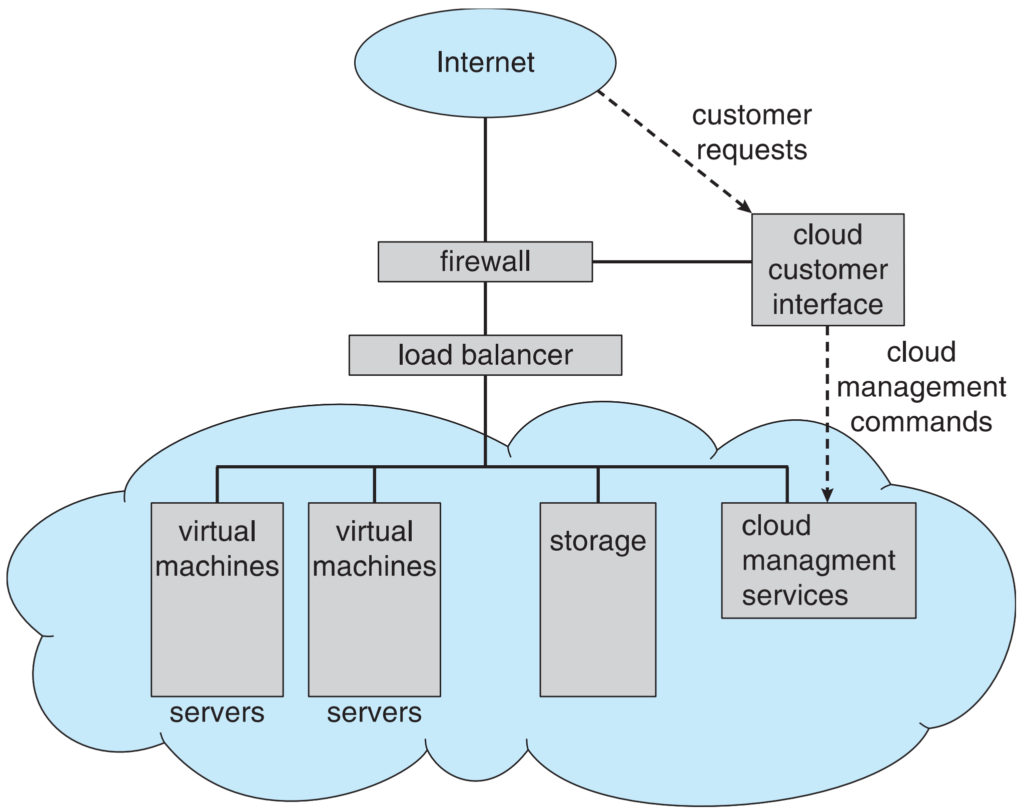


Computing Environments – Cloud Computing

- 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** 实时系统 如real-time Linux
  - 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

Free and Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary **closed-source** and **proprietary**
- Counter to the **copy protection** and **Digital Rights Management (DRM)** movement
- Started by **Free Software Foundation (FSF)**, which has "copyleft" **GNU Public License (GPL)**
  - Free software and open-source software are two different ideas championed by different groups of people
    - http://gnu.org/philosophy/open-source-misses-the-point.html/
- 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

The Study of Operating Systems

There has never been a more interesting time to study operating systems, and it has never been easier. The open-source movement has overtaken operating systems, causing many of them to be made available in both source and binary (executable) format. The list of operating systems available in both formats includes Linux, BUSD UNIX, Solaris, and part of macOS. The availability of source code allows us to study operating systems from the inside out. Questions that we could once answer only by looking at documentation or the behavior of an operating system we can now answer by examining the code itself.

Operating systems that are no longer commercially viable have been open-sourced as well, enabling us to study how systems operated in a time of fewer CPU, memory, and storage resources. An extensive but incomplete list of open-source operating-system projects is available from https://curlie.org/Computers/Software/Operating_Systems/Open_Source/

In addition, the rise of virtualization as a mainstream (and frequently free) computer function makes it possible to run many operating systems on top of one core system. For example, VMware (http://www.vmware.com) providesa free “player” for Windows on which hundreds of free “virtual appliances” can run. Virtualbox (http://www.virtualbox.com) provides a free, open-source virtual machine manager on many operating systems. Using such tools, students can try out hundreds of operating systems without dedicated hardware.

The advent of open-source operating systems has also made it easier to make the move from student to operating-system developer. With some knowledge, some effort, and an Internet connection, a student can even create a new operating-system distribution. Just a few years ago, it was difficult or impossible to get access to source code. Now, such access is limited only by how much interest, time, and disk space a student has.

Homework

- Exercises at the end of Chapter 1 (OS book)
  - 1.1, 1.3, 1.5, 1.6, 1.10, 1.11

End of Chapter 1