UNIVERSITY of SAN CARLOS
SCIENTIA • VIRTUS • DEVOTIO

DCISM
DEPARTMENT OF COMPUTER, INFORMATION SCIENCES AND MATHEMATICS

2

CS 3104
OPERATING SYSTEMS

CHAPTER 1

RESOURCE MANAGEMENT: IMPORTANT FACTS

- An **operating system** is a **resource manager**.

- **Resources** that **OS must manage**:
  - system's CPU
  - memory space
  - file-storage space
  - I/O devices

PROCESS MANAGEMENT

- **Process**:
  - It 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 (input)

- **Important Note:**
  - When the **process terminates**, the **OS will reclaim** any **reusable resources**.

PROCESS MANAGEMENT

- **Single-threaded process** has one **program counter** specifying location of the next instruction to execute
- **Process** executes instructions **sequentially, one at a time**, until completion
- **Multithreaded process** has **one program counter** per thread
- Typically, a **system has many processes** running concurrently on one or more CPUs.
  - **Operating-system processes** (**those that execute system code)**
  - **User processes** (**those that execute user code**)
  - **Concurrency** is done by **multiplexing the CPUs** among the **processes / threads**

PROCESS MANAGEMENT ACTIVITIES

- The OS is responsible for the following activities in connection with process management:
  - Creating and deleting both user and system processes
  - Scheduling processes and threads on the CPUs
  - 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 are **needed by the program must be in memory**

- **Memory management** determines:
  - what is in memory and
  - when optimizing CPU utilization and computer's response to users

- **Memory management activities:**
  - Keeping track of which parts of memory are currently being used and which process is using them
  - Deciding which processes (or parts of processes) 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**
  - OS 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
    - Creating and deleting directories to organize files
    - Supporting primitives for manipulating files and directories
    - Mapping files onto mass storage
    - Backing up files on stable (nonvolatile) storage media

MASS-STORAGE MANAGEMENT

- Usually, **these are 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

MASS-STORAGE MANAGEMENT

- **OS activities with secondary storage management:**
  - Mounting and unmounting
  - Free-space management
  - Storage allocation
  - Disk scheduling
  - Partitioning
  - Protection

- Some storage need not be fast
  - Tertiary storage includes magnetic tape, optical storage, and Blu-ray drives
  - Not crucial but still must be managed (by OS or applications)

CACHE MANAGEMENT

- **Caching:**
  - Important principle of computer systems
  - Performed at many levels in a computer **(in hardware, operating system, software)**

- Information in use is copied from slower to faster storage temporarily

- **Faster storage (cache)** is checked first to determine if information is there
  - If it is, information is used directly from the cache (fast)
  - If not, information is copied to cache and it is used there

- **Cache is smaller** than the storage being cached
  - Cache management is important in design problem
  - Cache size and replacement policy are also considered

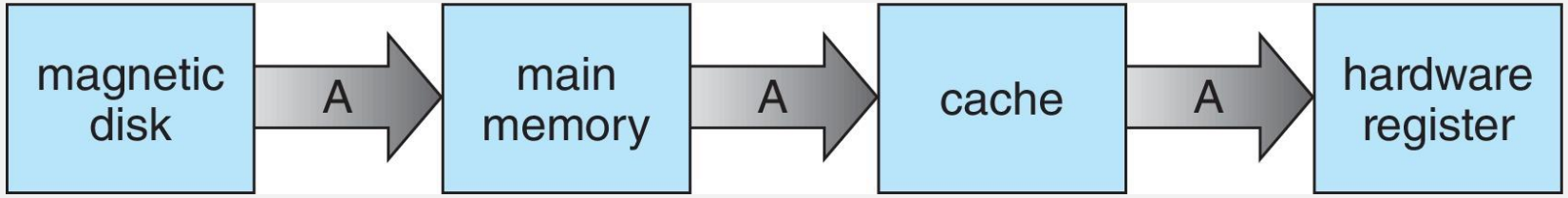
CHARACTERISTICS OF VARIOUS TYPES OF STORAGE

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	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 **is even more complex**
  - Several copies of a datum can exist
  - Various solutions are covered in Chapter 19

I/O SUBSYSTEM

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

- **Peculiarities of I/O devices are hidden** from the bulk of the operating system itself by the **I/O subsystem**

- **I/O subsystem** is 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**

SECURITY AND PROTECTION

- **Security:** defense of the system against internal and external attacks
  - Huge range: denial-of-service, worms, viruses, identity theft, theft of service

- **Protection:** any mechanism for controlling access of processes or users to resources defined by the OS

- **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 management, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

VIRTUALIZATION

- The **creation of a virtual (rather than actual) version of something**, such as an operating system (OS), a server, a desktop, a storage device or network resources.
- A **technology** that **allows us to abstract the hardware of a single computer** into **several different execution environments**
- Creating the illusion that each separate environment is running on its own private computer
- Allows operating systems to run as applications within other Operating Systems
- Vast and growing industry

VIRTUALIZATION

- **Virtualization software is one member of a class that also includes emulation**
  - **Emulation** is **used when source CPU type is different from target type**
    - (i.e., PowerPC to Intel x86)
    - generally, it is the slowest method
    - emulated code may run much more slowly than the native code

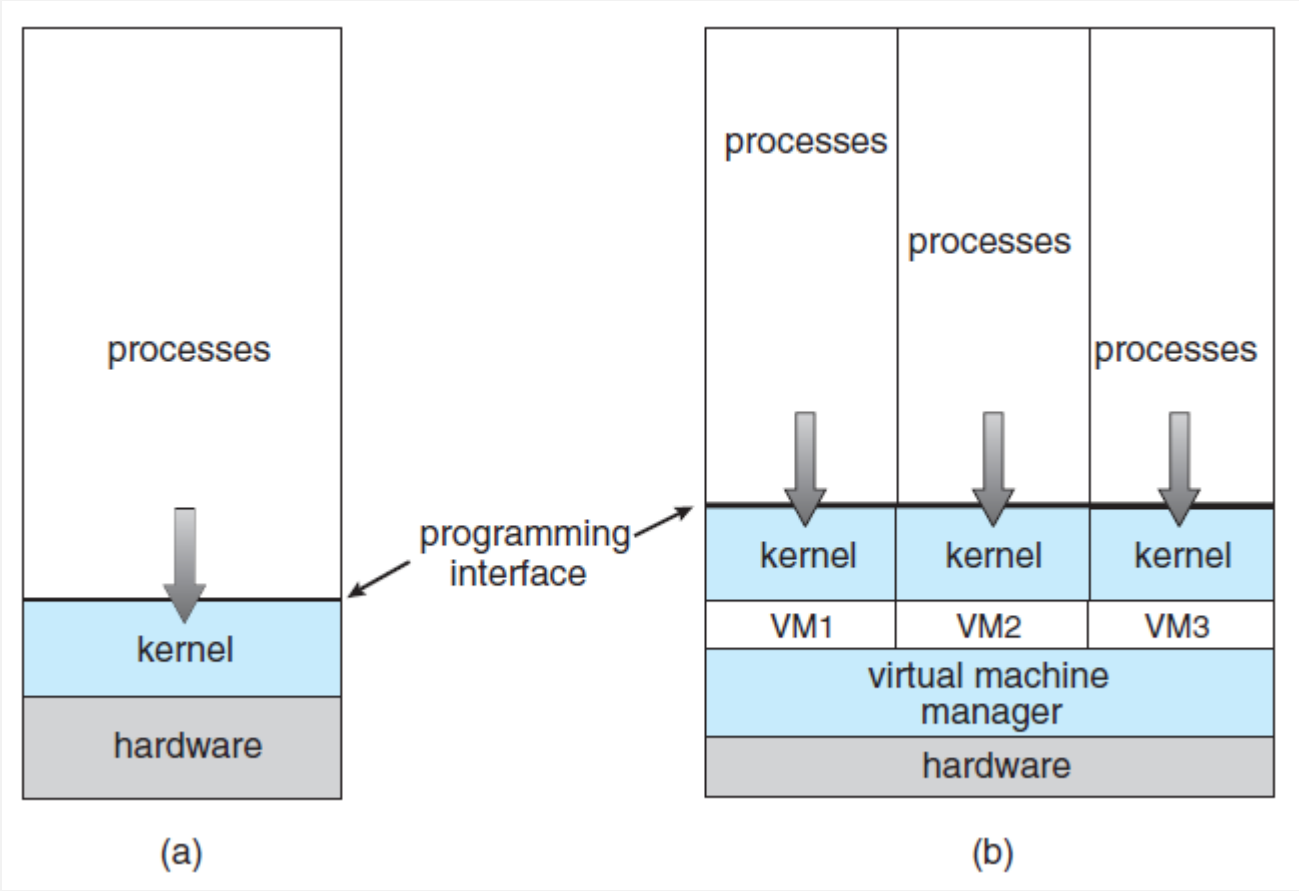
- **Virtualization: OS natively compiled for CPU**, running **guest** Operating Systems also natively compiled
  - Consider **VMware running WinXP guests**, **each running applications**, **all on native WinXP host OS**
  - **VMM** (**V**irtual **M**achine **M**anager) provides **virtualization services**

VIRTUALIZATION

- **Use cases** that **involve laptops and desktops running multiple Operating Systems for exploration or compatibility**
  - **Apple laptop** running **Mac OS X host**, **Windows as a guest**
  - **Developing apps** for multiple Operating Systems 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



A computer running (a) a single operating system and (b) three virtual machines.

DISTRIBUTED SYSTEMS

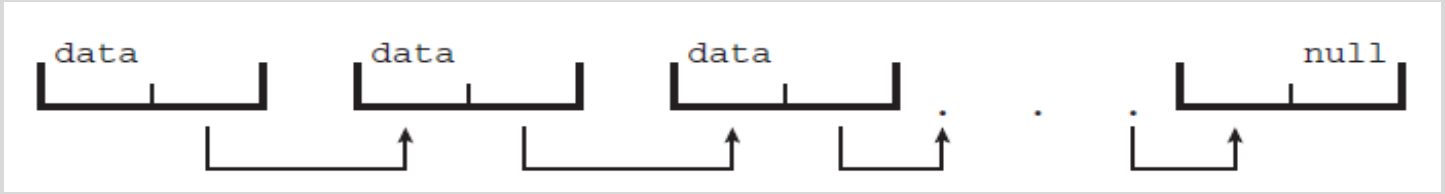
- **Distributed computing**: a field of **computer science** that **studies distributed systems**

- **Distributed Systems:**
  - **Collection** of **physically separate, possibly heterogeneous computer systems networked together**
    - **Network** is a **communication path between 2 or more systems**
      - **TCP/IP** is the most common network protocol used
        - **Local Area Network (LAN)**
        - **Wide Area Network (WAN)**
        - **Metropolitan Area Network (MAN)**
        - **Personal Area Network (PAN)**

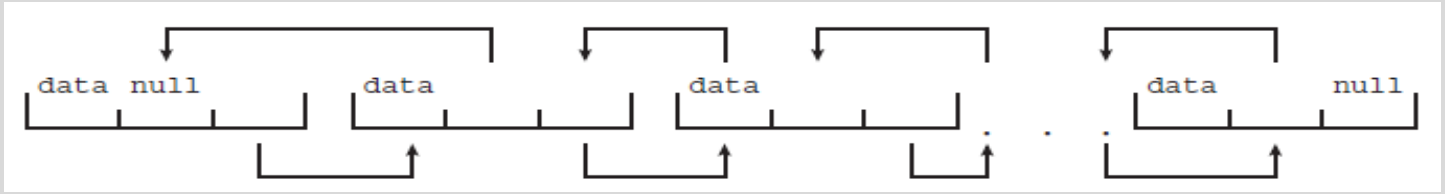
- **Network Operating System:** provides features between systems across network
  - Communication scheme allows systems to exchange messages
  - Illusion of a single system

KERNEL DATA STRUCTURES

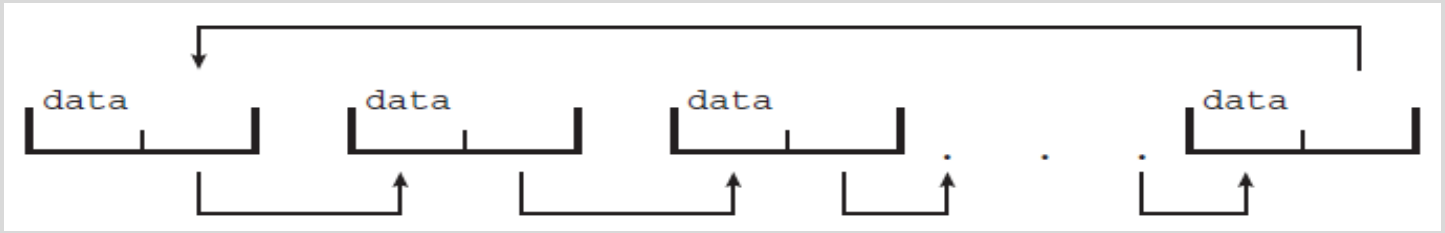
- **Many similar to standard programming data structures:**
  - **Singly linked list**



  - **Doubly linked list**

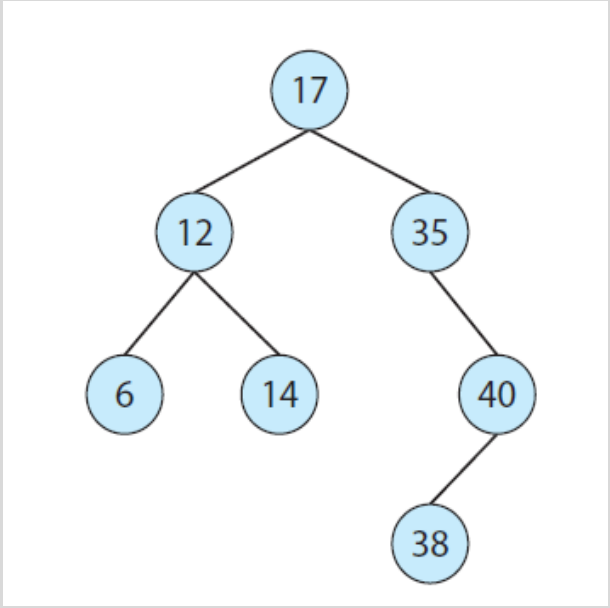


  - **Circularly linked list**



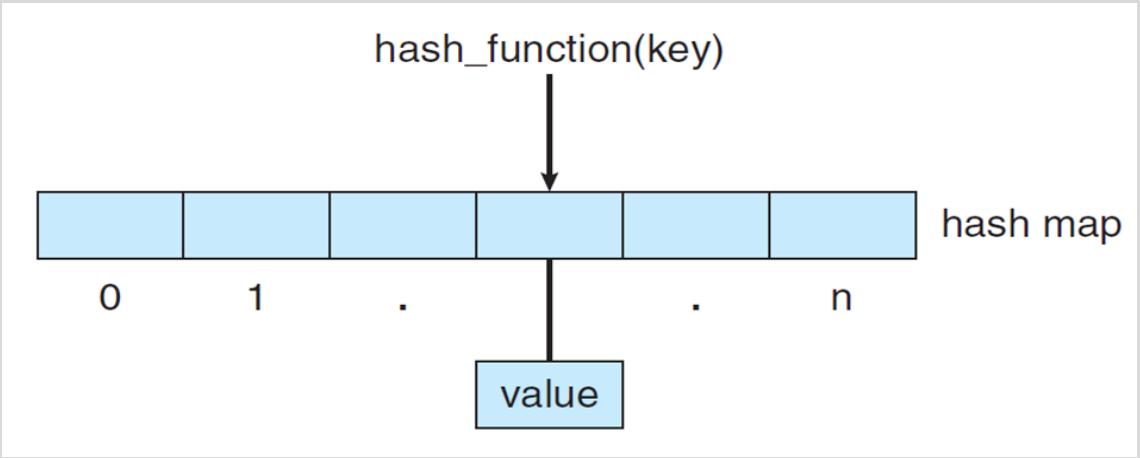
KERNEL DATA STRUCTURES

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



KERNEL DATA STRUCTURES

- **Hash function** can create a **hash map**
- **Hash Map:**
  - which **associates** (or **maps**) [**key:value**] **pairs** using **a hash function**



- **Hash function:**
  - **takes data as its input, performs a numeric operation on the data**, and **returns a numeric value**
  - this **numeric value** can then be **used as an index into a table** (typically an array) **to quickly retrieve the data**

- **Example:**
  Suppose that a user name is mapped to a password
  - **Password authentication** then proceeds as follows: **a user enters her user name and password**
  - **Hash function** is **applied to the user name**, which is then **used to retrieve the password**
  - The **retrieved password** is then **compared with the password entered by the user** for **authentication**

KERNEL DATA STRUCTURES

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

- **Example:**
  Suppose we have several resources; availability of each resource is **indicated by the value of a binary digit**: **0** means **resource is available**, while **1** indicates that **it is unavailable (or vice versa)**

- Consider the **bitmap** shown below:
  001011101
  - Resources 2, 4, 5, 6, and 8 are unavailable
  - Resources 0, 1, 3, and 7 are available

- **Bitmaps:** commonly used when there is a need to represent the availability of a large number of resources.
- **Example:** Disk drives
  - Medium-sized disk drive is divided into several thousand individual units (**disk blocks**)
  - A **bitmap** can be used to **indicate the availability of each disk block**

LINUX KERNEL DATA STRUCTURES

- The **data structures** used in the Linux kernel are available in the kernel source code.
- The ***include*** file **<linux/list.h>** provides details of the linked-list data structure used throughout the kernel.
- A **queue** in Linux is known as a **kfifo**, and its implementation can be found in the **kfifo.c** file in the **kernel** directory of the source code.
- Linux also provides a balanced binary search tree implementation using ***red-black trees***.
- **Details** can be found in the include file **<linux/rbtree.h>**.

COMPUTING ENVIRONMENTS: TRADITIONAL

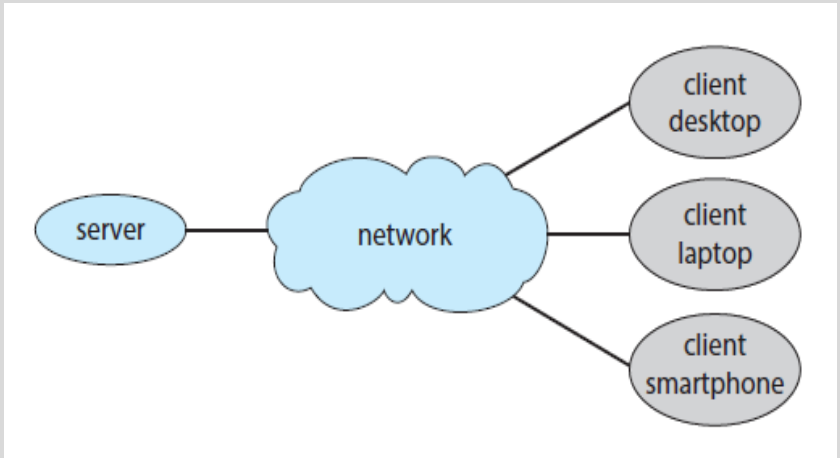
- **Stand-alone** general-purpose machines
- But **blurred**, as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers** (**thin clients**) are like Web terminals
- **Mobile computers** interconnect via **wireless networks**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

COMPUTING ENVIRONMENTS: MOBILE

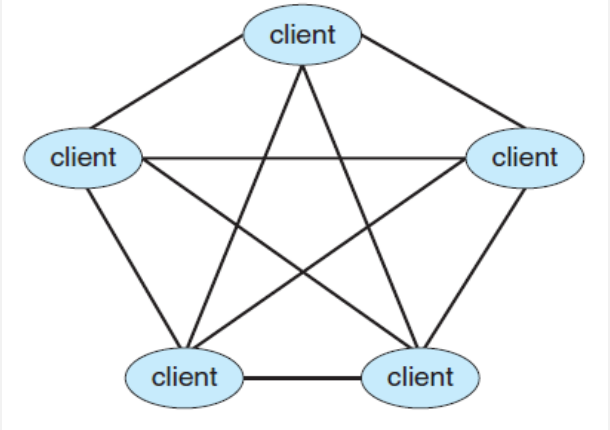
- Handheld smartphones, tablets, etc.
- What is the functional difference between them and a “traditional” laptop?
- Extra feature – more OS features (GPS, gyroscope)
- Allows new types of apps like ***augmented reality***
- Use IEEE 802.11 wireless, or cellular data networks for connectivity
- Leaders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

COMPUTING ENVIRONMENTS: CLIENT-SERVER

- **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: CLOUD COMPUTING

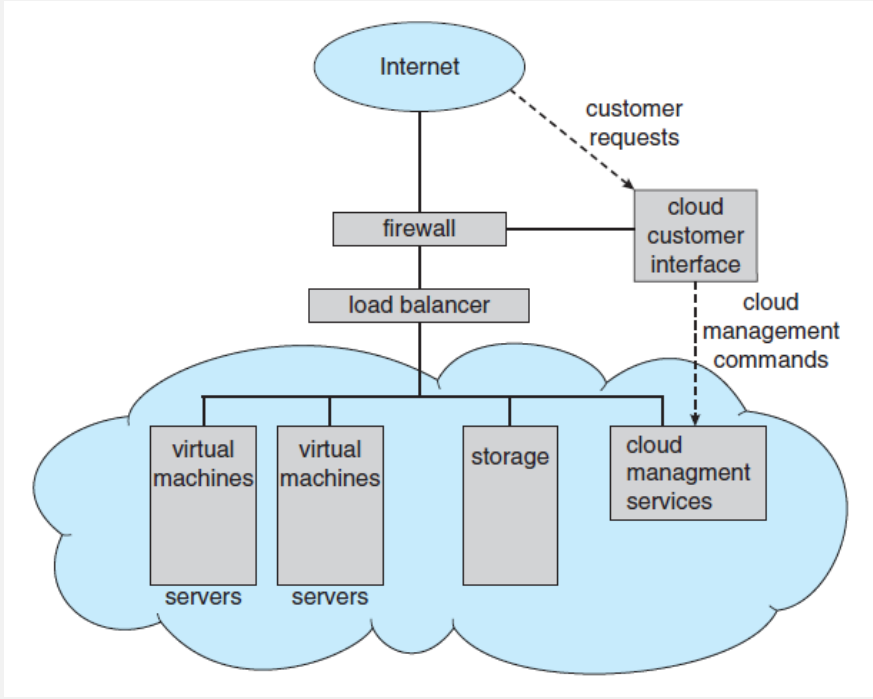
- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for its functionality
- Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay is based on usage

COMPUTING ENVIRONMENTS: CLOUD COMPUTING

- **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
  - **S**oftware **a**s **a** **S**ervice (**SaaS**): one or more applications available via the Internet (i.e., word processor)
  - **P**latform **a**s **a** **S**ervice (**PaaS**): software stack ready for application use via the Internet (i.e., a database server)
  - **I**nfrastructure **a**s **a** **S**ervice (**IaaS**): servers or storage available over Internet (i.e., storage available for backup use)

COMPUTING ENVIRONMENTS: CLOUD COMPUTING

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



REAL-TIME EMBEDDED SYSTEMS

- **Real-time embedded systems** are the 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 Operating System
  - 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 **D**igital **R**ights **M**anagement **(DRM)** movement

- Started by **F**ree **S**oftware **F**oundation **(FSF)**, which has **"copyleft" GNU Public License (GPL)**
  - Free software and open-source software are two different ideas championed by different groups of people

FREE AND OPEN-SOURCE OPERATING SYSTEMS

- 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 - https://www.virtualbox.org)
  - Used to run guest operating systems for exploration

UNIVERSITY of SAN CARLOS
SCIENTIA • VIRTUS • DEVOTIO

DCISM
DEPARTMENT OF COMPUTER, INFORMATION SCIENCES AND MATHEMATICS

2

CS 3104
OPERATING SYSTEMS

*** END ***

THANK YOU!