Introduction to Computer Science:

OS

May 2020

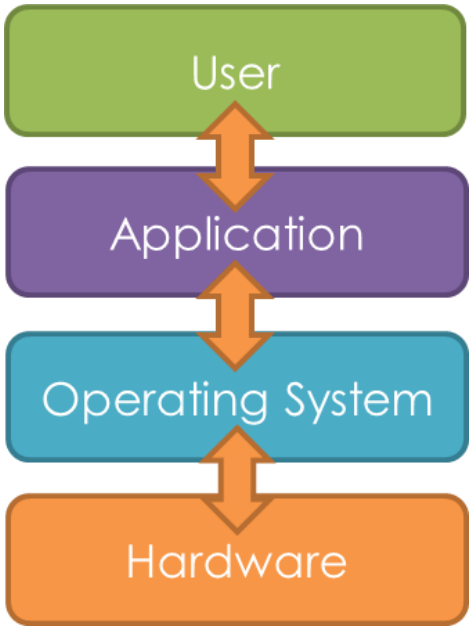
Honguk Woo

without Operating System

with Operating System

Intro. to Operating Systems

- An operating system performs as an intermediary between the user and the hardware, functioning as a virtual computer that makes the use of the hardware more convenient and/or efficient

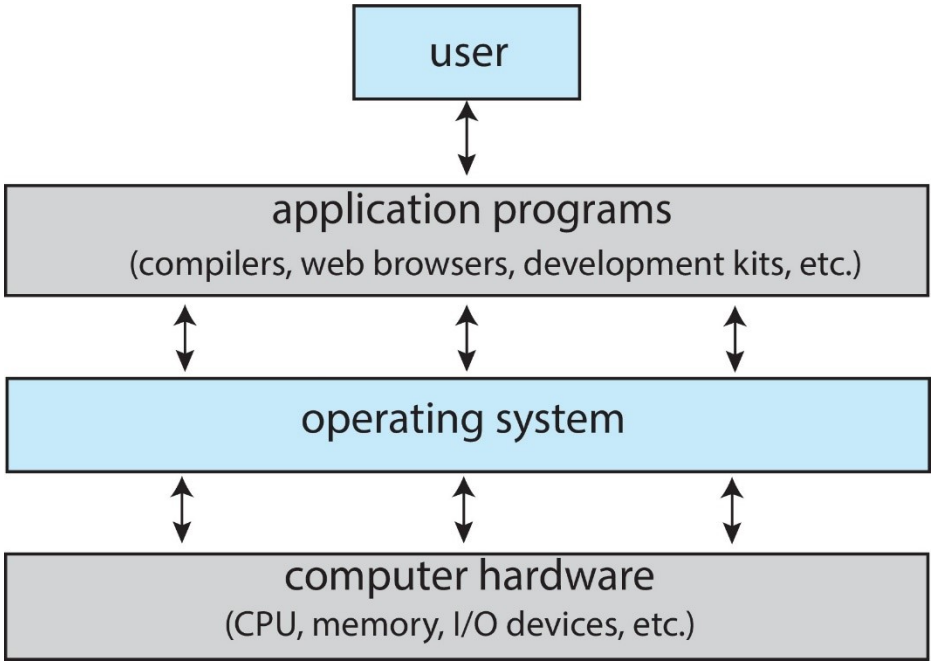


Computer System Structure

- Computer system can be divided into four components:
  - Hardware : provides basic computing resources
    - CPU, memory, I/O devices …
  - 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 …
  - Operating system : controls and coordinates use of hardware among various applications and users
    - MS Windows, Linux, Unix, macOS …
  - Users
    - People, machines, other computers

What is an Operating System

- Abstract view of the components of Computer System



What Operating Systems Do

- The operating system can be observed from the point of view of the user or the system.
- User view
  - The user's view depends on the system interface.
  - The different types of user view experiences :
    - Personal computer : the operating system is designed to make the interaction easy. There is no need for the operating system to worry about resource utilization.
    - Mainframe : the operating system is concerned with resource utilization; makes sure that all the resources are divided properly.
    - Network computing : if the user is sitting on a workstation connected to other workstations through networks, the operating system needs to focus on both individual usage of resources and sharing though the network.

What Operating Systems Do

- System view
  - Resource allocator : there are many resources such as CPU time, memory space, file storage space, I/O devices etc. that are required by processes for execution. The computer relies on the operating system to allocate, manage, and control the resources.
  - Control program : it manages all the processes and I/O devices so that the computer system works smoothly and there are no errors.

Defining Operating Systems

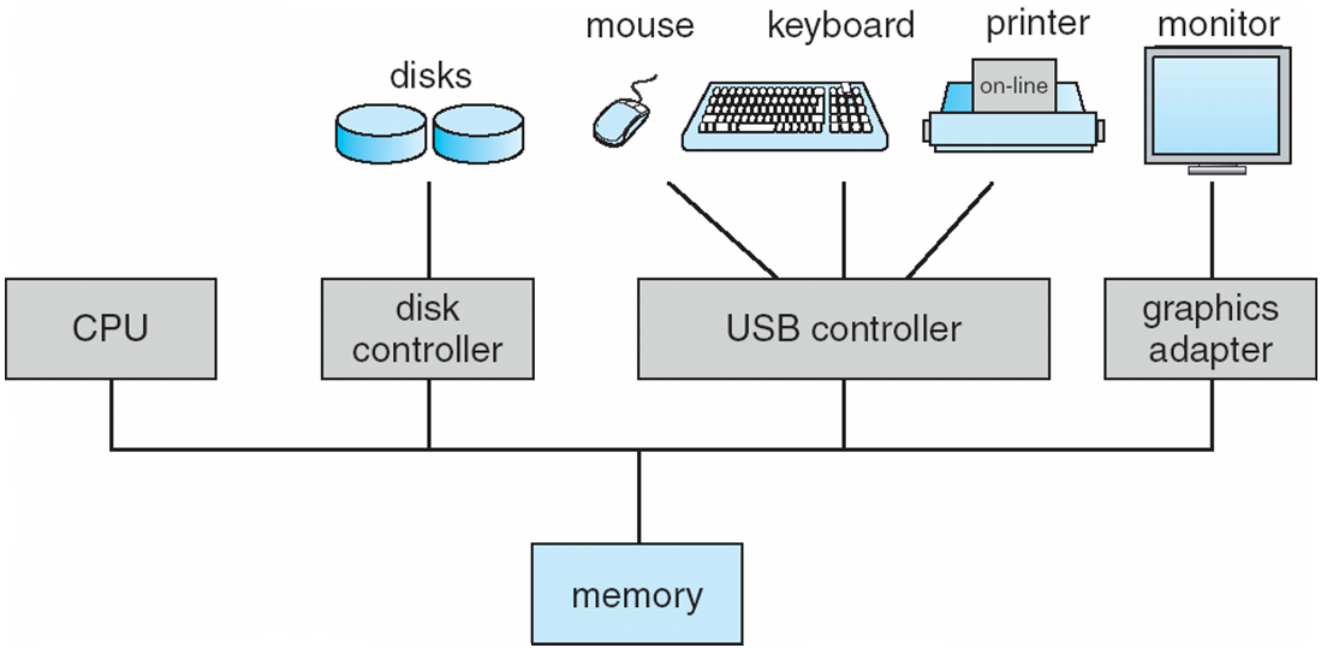
- No universally accepted definition of an operating system
  - “Everything a vendor ships when you order an operating system” is a good approximation, but varies wildly.
- The operating system (OS) covers many roles and functions:
  - Various designs and uses of computers : present in e.g., toasters, ships, spacecraft, game machines, TVs and industrial control systems
  - Computers evolved : from fixed-purpose computers for military uses to more general purpose; so needing resource management and program control
- Definition of operating systems : resource allocator, control program
  - Manages all resources; decides between conflicting requests for efficient and fair resource use
  - Controls execution of programs to prevent errors and improper use of the computer
- When computer science professionals refer to the operating system, they mean *kernel*.

Defining Operating Systems (Cont.)

- **Kernel :** core part of an operating system
  - “The one program running at all times on the computer”
- Everything else is either
  - a system program (ships with the operating system, but not part of the kernel) , or
  - an application program, all programs not associated with the operating system

Computer System Organization

- A general purpose computer system consists of one or more CPUs and device controllers connected through common bus providing access to shared memory

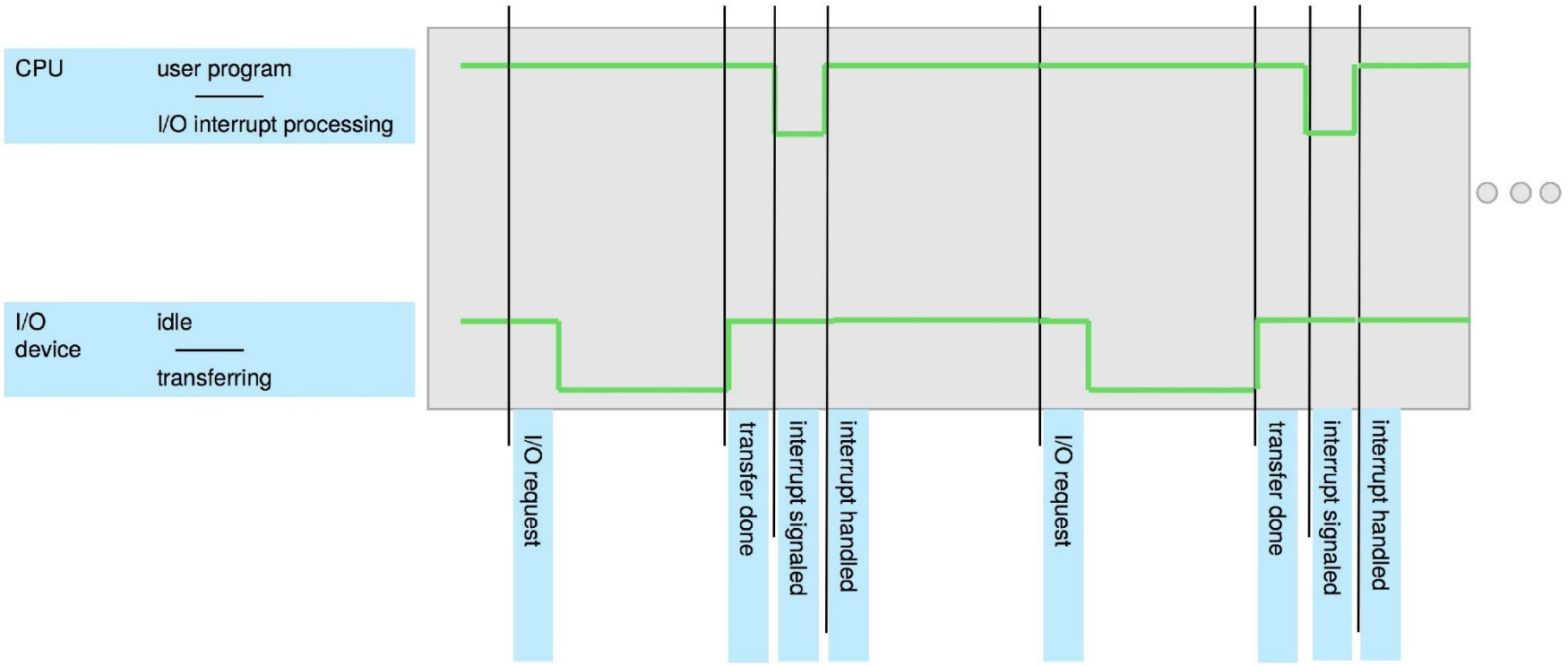


Computer System Operation

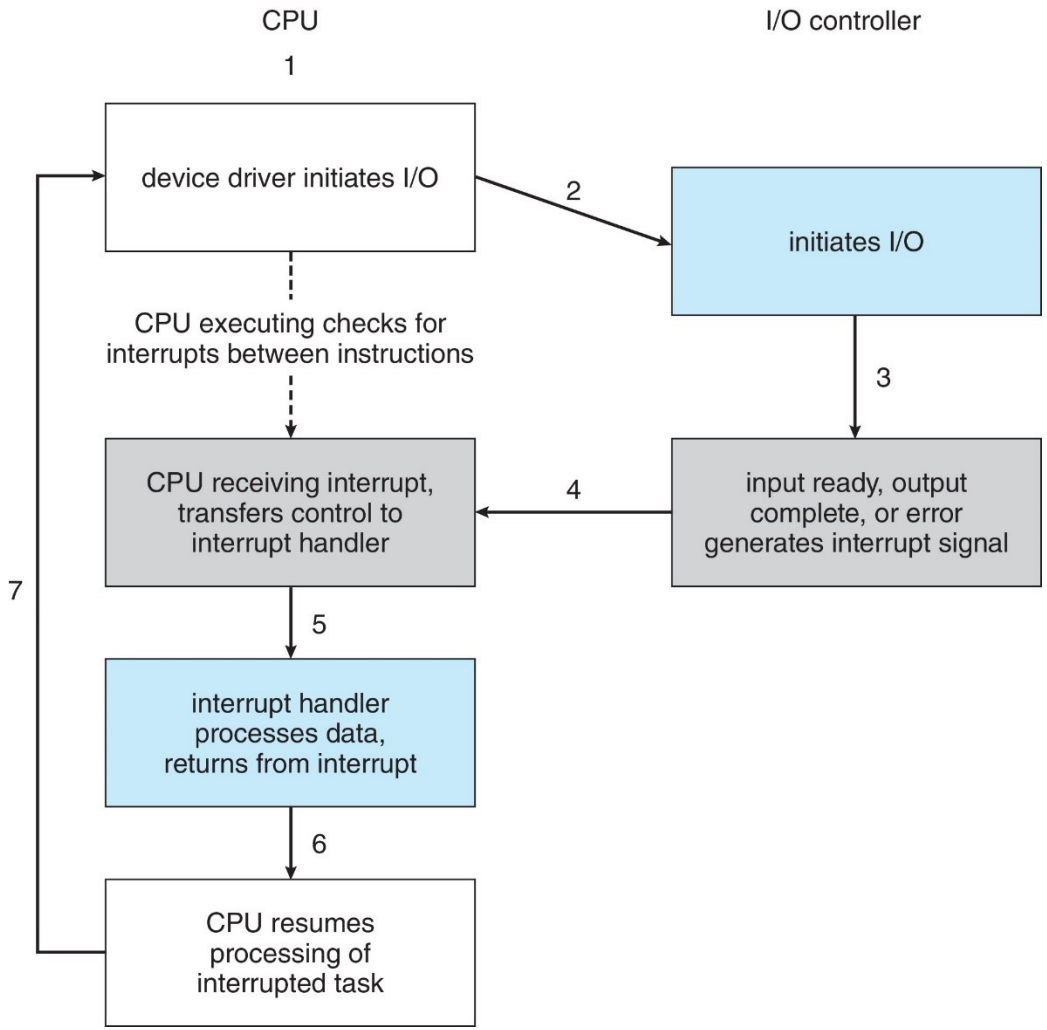
- How do the devices communicate with the operating system ?
  - I/O devices and the CPU can execute concurrently
  - Each device controller is in charge of a particular device type
  - Each device controller has a local buffer

  - When performing I/O, a device controller transfers data from the device to the local buffer
  - Once the transfer of data is complete, the device controller informs the device driver (in OS) that it has finished its operation by causing an interrupt
    - Alerting the CPU to events that require attention

Interrupt Timeline



Interrupt driven I/O cycle



Interrupts

- Interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the addresses of all the service routines
  - Separate segments of code determine what action should be taken for each type of interrupt
- 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

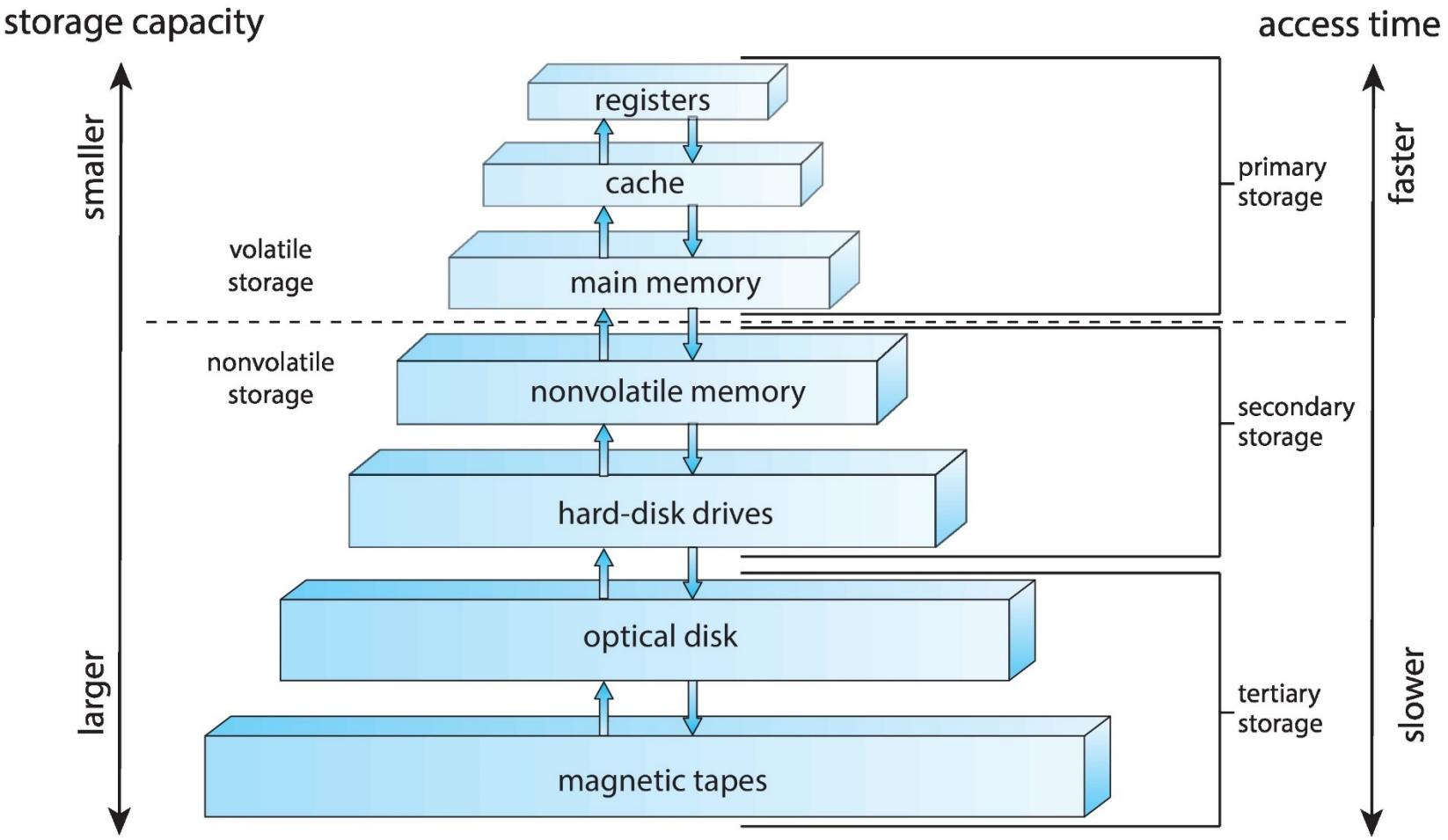
Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - Random access, Typically volatile
  - Byte-addressable : array of bytes where its byte its own address
- 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 (SSD) – faster than hard disks, nonvolatile
    - 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

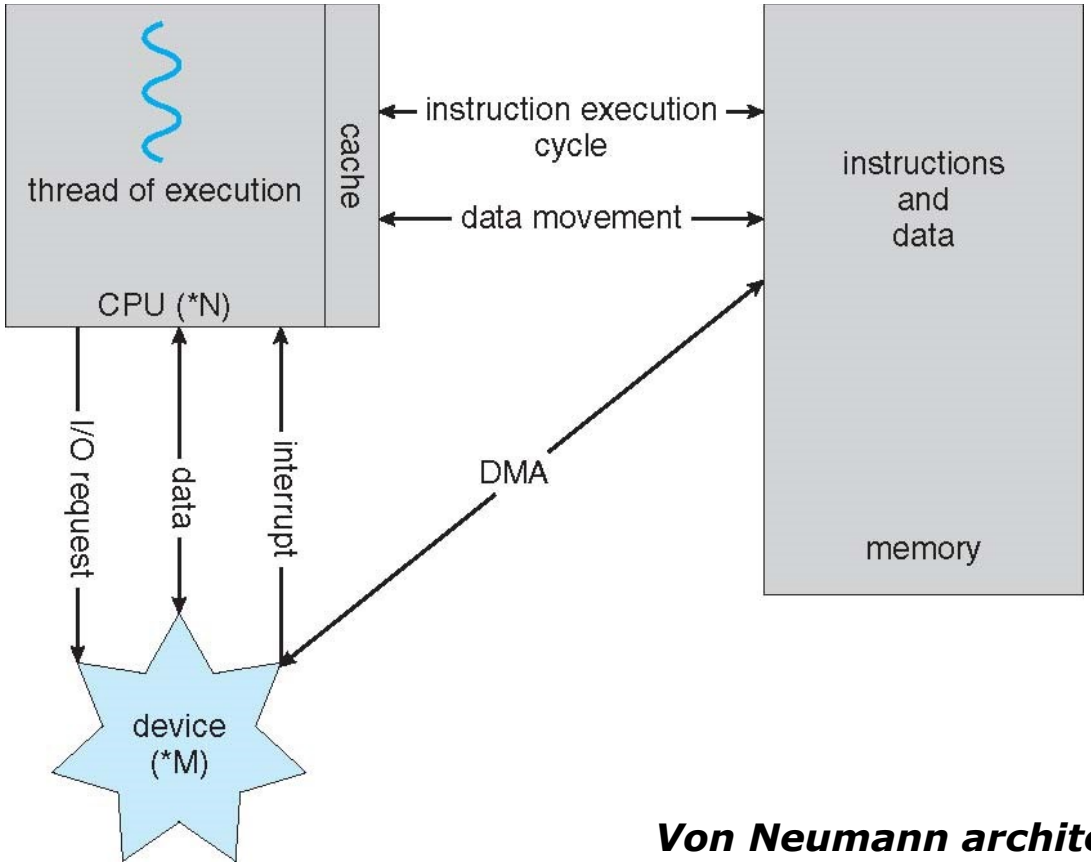
Notation Review

- **kilobyte**, or **KB**, is 1,024 bytes
- **megabyte**, or **MB**, is $1{,}024^2$ bytes
- **gigabyte**, or **GB**, is $1{,}024^3$ bytes
- **terabyte**, or **TB**, is $1{,}024^4$ bytes
- **petabyte**, or **PB**, is $1{,}024^5$ bytes

I/O Structure

- Interrupt-driven I/O
  - Some forms of I/O require that data be moved one byte at a time between the device controller and the main memory
  - When doing I/O this way, the controller has to interrupt the CPU after transferring every single byte. That is fine for devices that are inherently slow, like keyboards
  - But, it can produce high overhead when used for bulk data movement
- For devices like hard drives with the ability to transfer data at high speed, computing systems utilize direct memory access (**DMA**)

How a Modern Computer Works



Von Neumann architecture

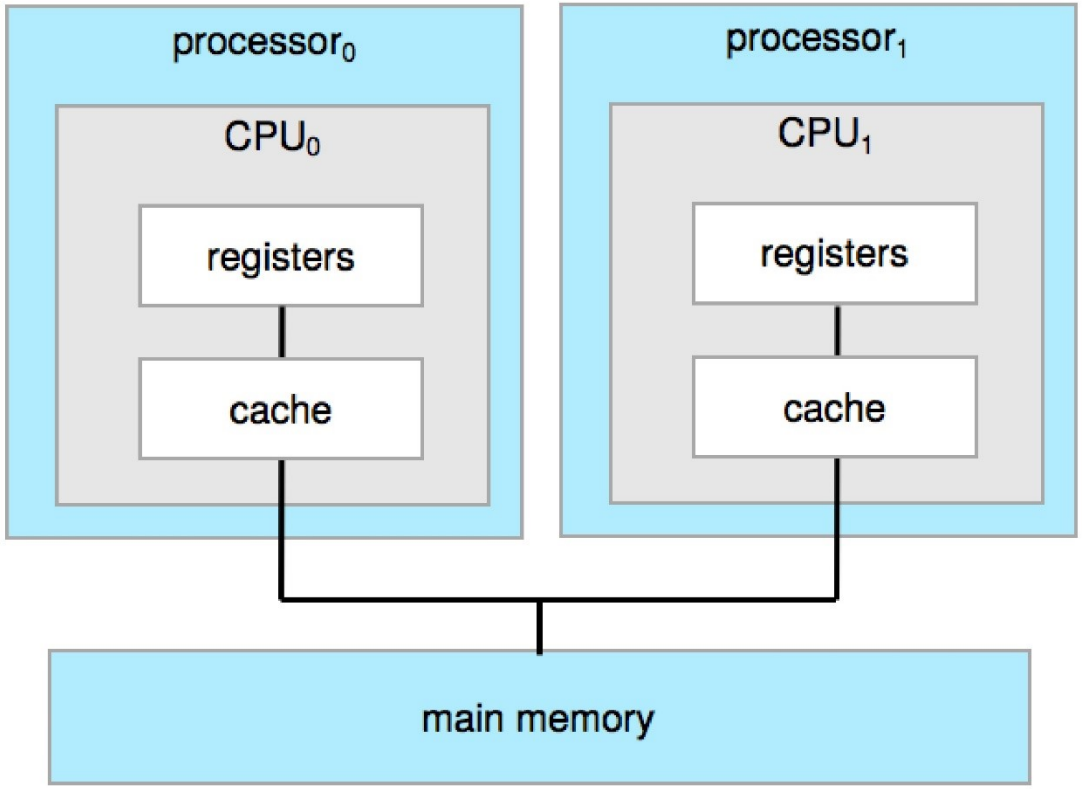
Direct Memory Access (DMA) Structure

- Used for high-speed I/O devices that are 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

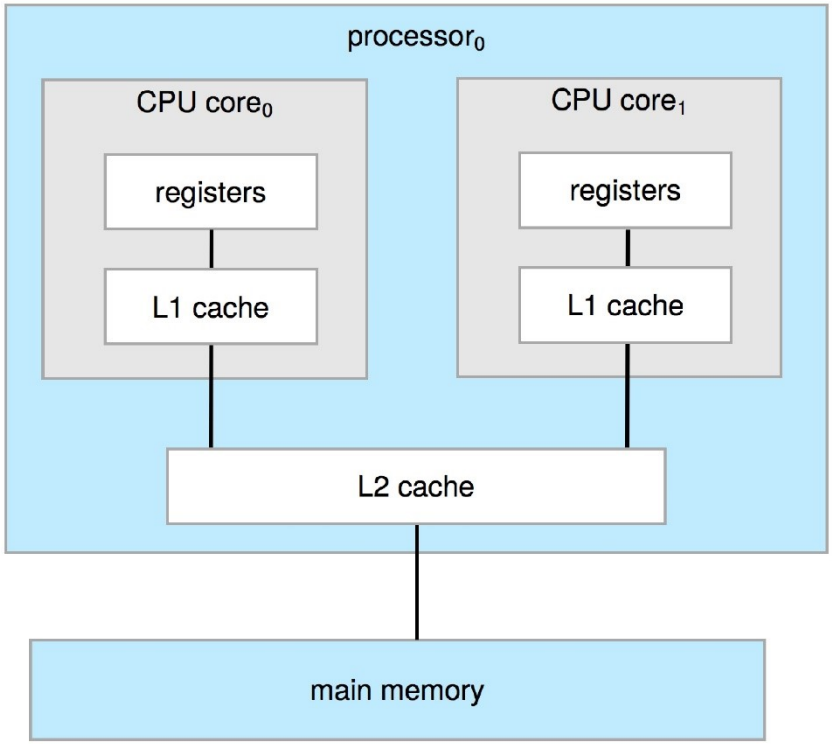
Computer System Architecture

- Multiprocessors systems growing in use and importance
  - Also known as **parallel systems;** Advantages include:
    - Increased throughput
    - Economy of scale
    - Increased reliability – graceful degradation or fault tolerance
  - Two types:
    - Asymmetric Multiprocessing – each processor is assigned a specific task.
    - Symmetric Multiprocessing – each processor performs all tasks

Symmetric Multiprocessing Architecture

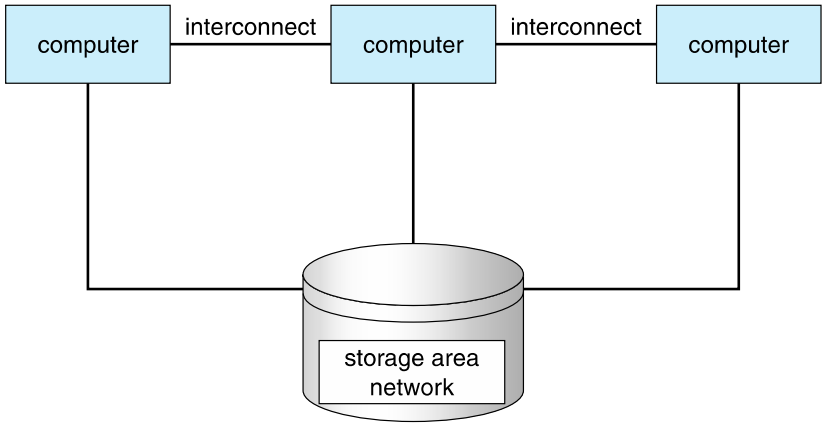


Dual-Core Design



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
  - 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



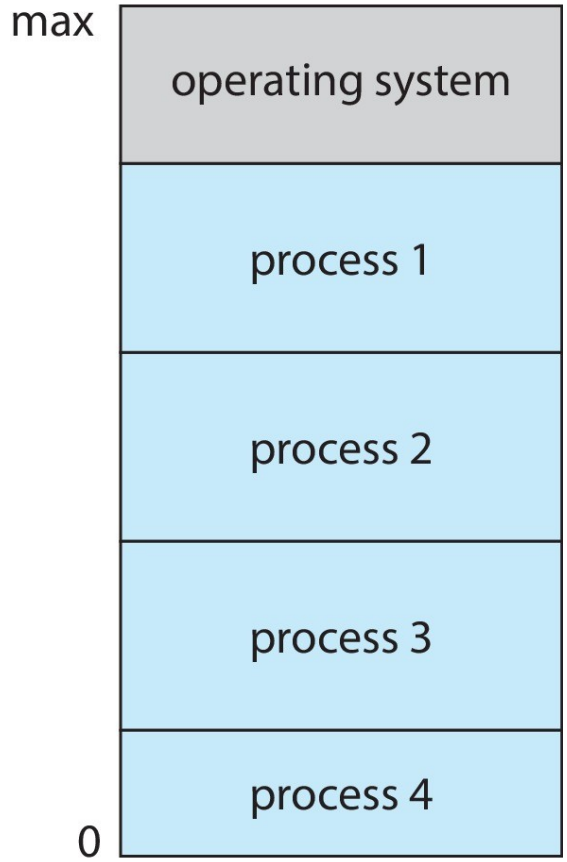
Operating System Operations

- OS provides the environment where programs are executed; Internally, OSes vary greatly; but share common functions.
- For example, consider how to start a computer :
  - Bootstrap program – simple code to initialize the system, load the kernel
  - Kernel load; Starts system daemons (e.g., systemd; services provided outside of the kernel)
  - Kernel is signaled by interrupt (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

- One of most important aspects of OSes is to run multiple programs
- Multiprogramming (Batch system) needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes multiple processes so CPU always has one to execute
  - A subset of total processes in system is kept in memory; One process is selected and run
  - When it has to wait (for I/O for example), OS switches to another process
- Timesharing (multitasking) is logical extension in which CPU switches processes so frequently that users can interact with each job while it is running, creating interactive computing
  - Response time should be < 1 second
  - If several processes ready to run at the same time ⇨ CPU scheduling
  - If processes don't fit in memory, swapping moves them in and out to run
  - Virtual memory allows execution of processes not completely in memory

Memory Layout for Multiprogrammed System

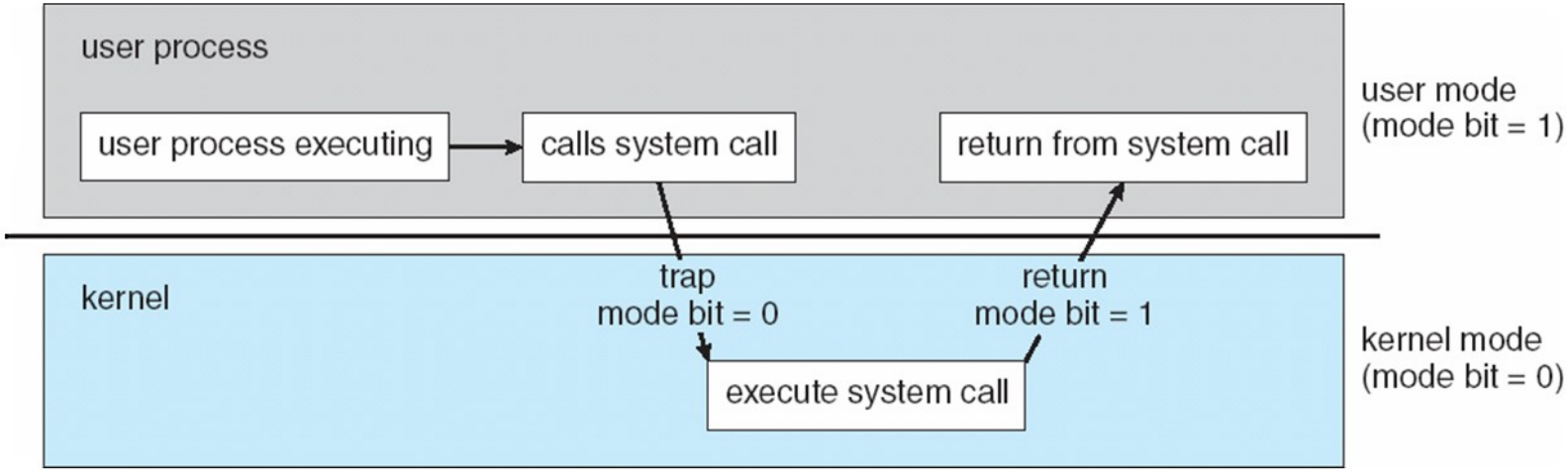


Dual mode

- OS and user programs share the computer resources, so some needs to ensure that an incorrect program cannot cause other programs to execute incorrectly

- Dual-mode operation allows OS to protect itself and other system components
  - User mode and kernel mode
  - Mode bit is 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

Transition from User to Kernel Mode



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
- 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

Process Management Activities

- The operating system has to
  - create and delete processes,
  - schedule processes (and threads) on CPUs,
  - suspend and resume processes,
  - provide mechanisms for process synchronization, and
  - provide mechanisms for process communication

Memory Management Activities

- The operating system has to
  - keep track of which parts of memory are currently being used, and which processes are using them
  - allocate and deallocate memory space as needed
  - decide which processes (or parts of processes) and data to copy into or out of memory

File System Activities

- The operating system has to
  - create and delete files,
  - create and delete directories (folders),
  - support primitives for manipulating files and directories,
  - map files onto mass storage, and
  - back up files on stable (nonvolatile) storage media.

Mass Storage Management Activities

- The operating system has to
  - mount and unmount devices,
  - manage free space,
  - allocate storage,
  - schedule disk accesses,
  - perform partitioning, and
  - provide protection

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
- Desktop PCs
  - e.g., a web browser is composed of multiple processes

Computing Environments - Mobile

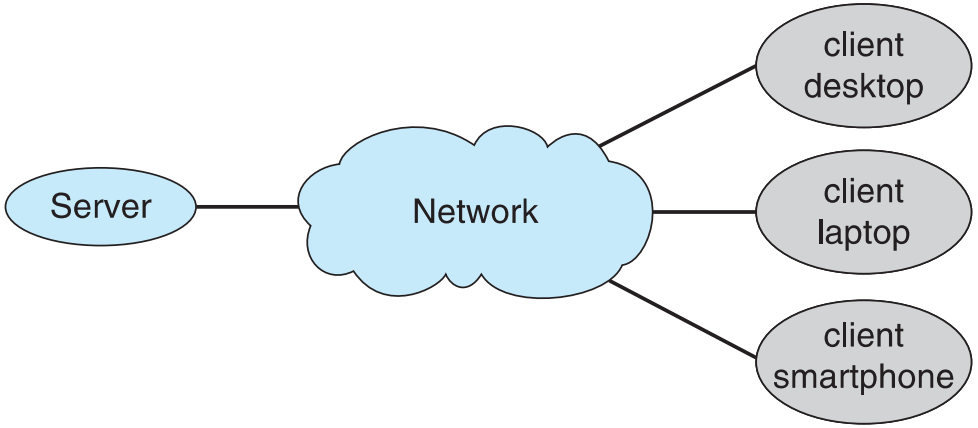
- Handheld smartphones, tablets, etc
- What is the functional difference between them and a “traditional” laptop?
- Extra feature – more OS features (GPS, gyroscope)
- Allows new types of apps like *augmented reality*
- Use IEEE 802.11 wireless, or cellular data networks for connectivity
- Leaders are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

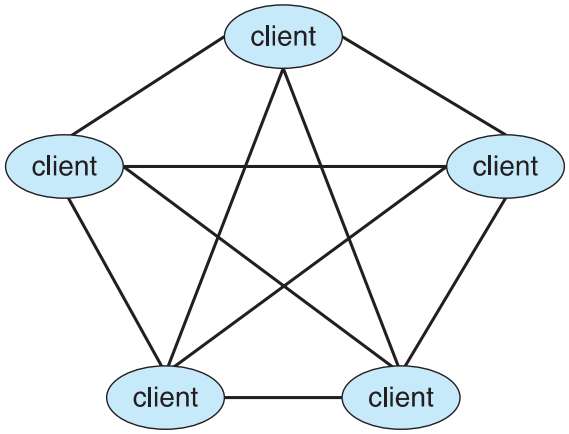
Computing Environments – Client-Server

- Client-Server Computing
  - 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



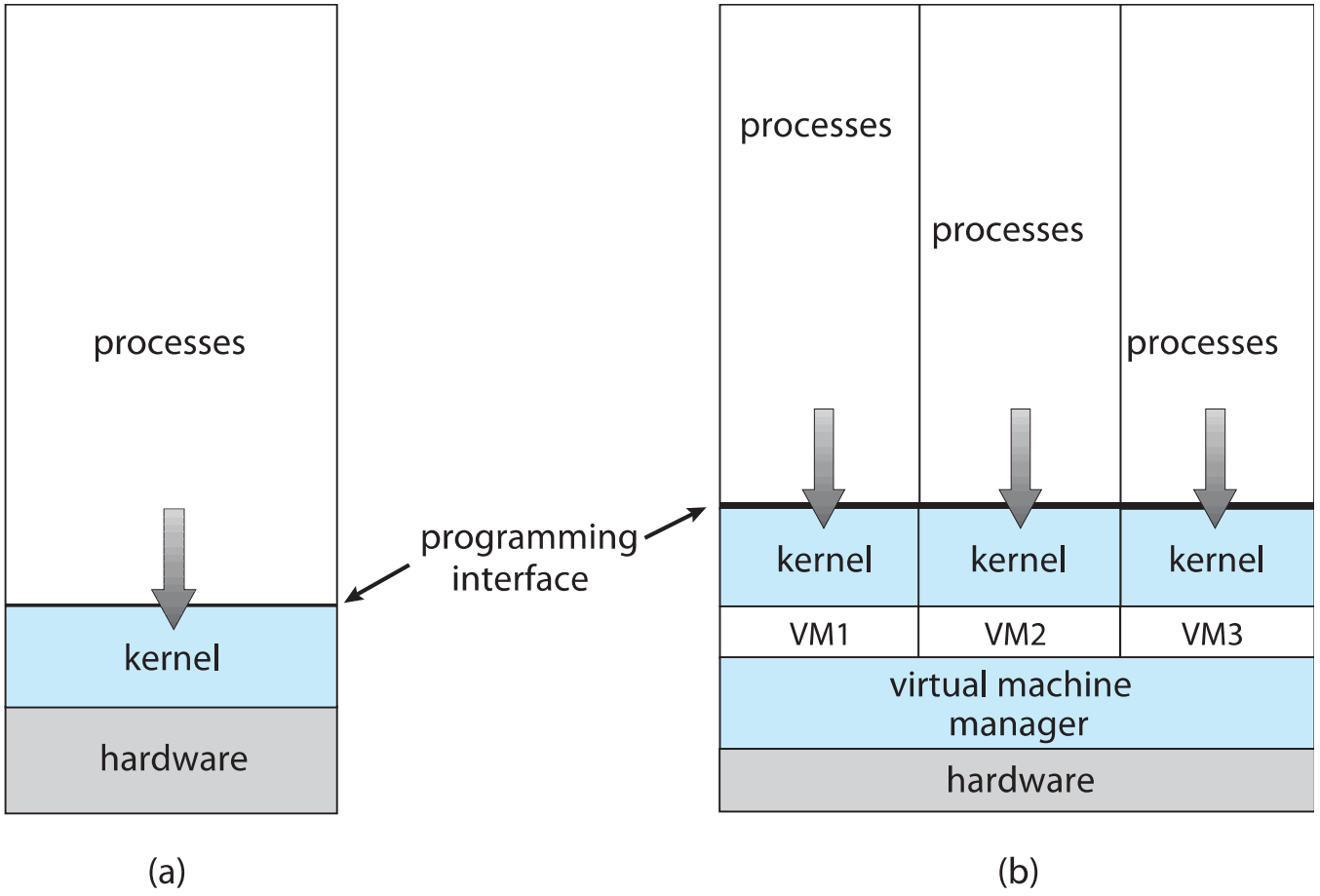
Computing Environments - Virtualization

- 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 Windows guest OS;
    each windows guest OS supports to run applications, all on native Windows 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

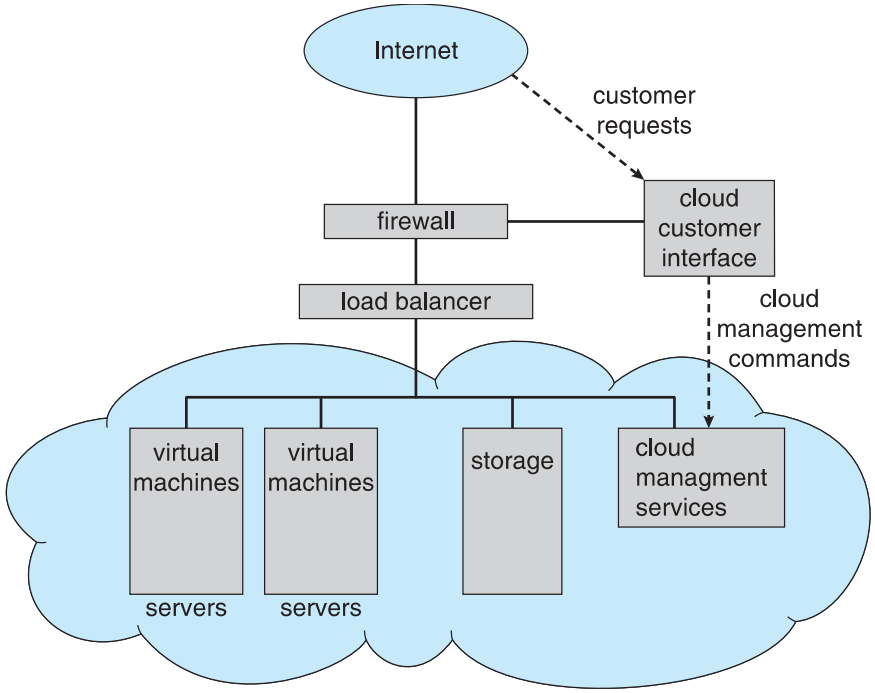


Computing Environments – Cloud Computing

- 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
- 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

Summary : Operating System

- Operating system : System software that
  - **manages** computer resources, such as memory and input/output devices
  - **provides** an interface through which a human can interact with the computer
  - **allows** an application program to interact with these other system resources

- *An operating system implements a virtual machine that is easier and safer to program and use than the raw hardware*
  - The various roles of an operating system generally revolve around the idea of "sharing nicely"
  - **Coordinator** : an operating system manages resources, and these resources are often shared in one way or another among programs that want to use them

Discussion :
which OSes do you use ?

- Server OS
  - **Linux**, Windows
- Mobile OS
  - **Android**, **iOS**
- Desktop OS
  - **Windows**, Mac