Chapter 1: Introduction

Operating system concepts / Abraham Silberschatz,

Edited by : Dr. Mustafa Shiple

What Does the Term Operating System Mean?

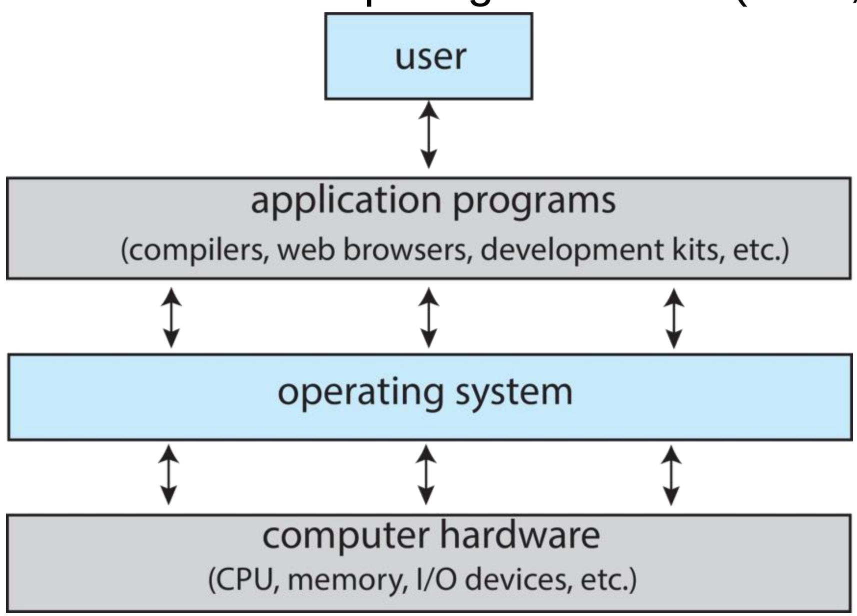
- An operating system is “fill in the blanks”
- What about:
  - Car
  - Airplane
  - Printer
  - Washing Machine
  - Toaster
  - Compiler
  - Etc.

What is an Operating System?

- A program that acts as an **intermediary** between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Computer System Structure

- Computer system can be divided into four components:
  - **Users (** People, machines, other computers)
  - **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.
  - **Operating system** - Controls[1] and coordinates[2] use of hardware among various *applications* and *users*
  - **Hardware** – provides basic computing resources (CPU, memory, I/O devices)



What Operating Systems Do

- Depends on the point of view
- Users View, **ease of use** and **good performance**
  - Don't care about **resource utilization**
- System view, Operating system is a **resource allocator (**CPU time, memory space, storage space, I/O devices, and so on) and **control program** (prevent errors and improper use) ; 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

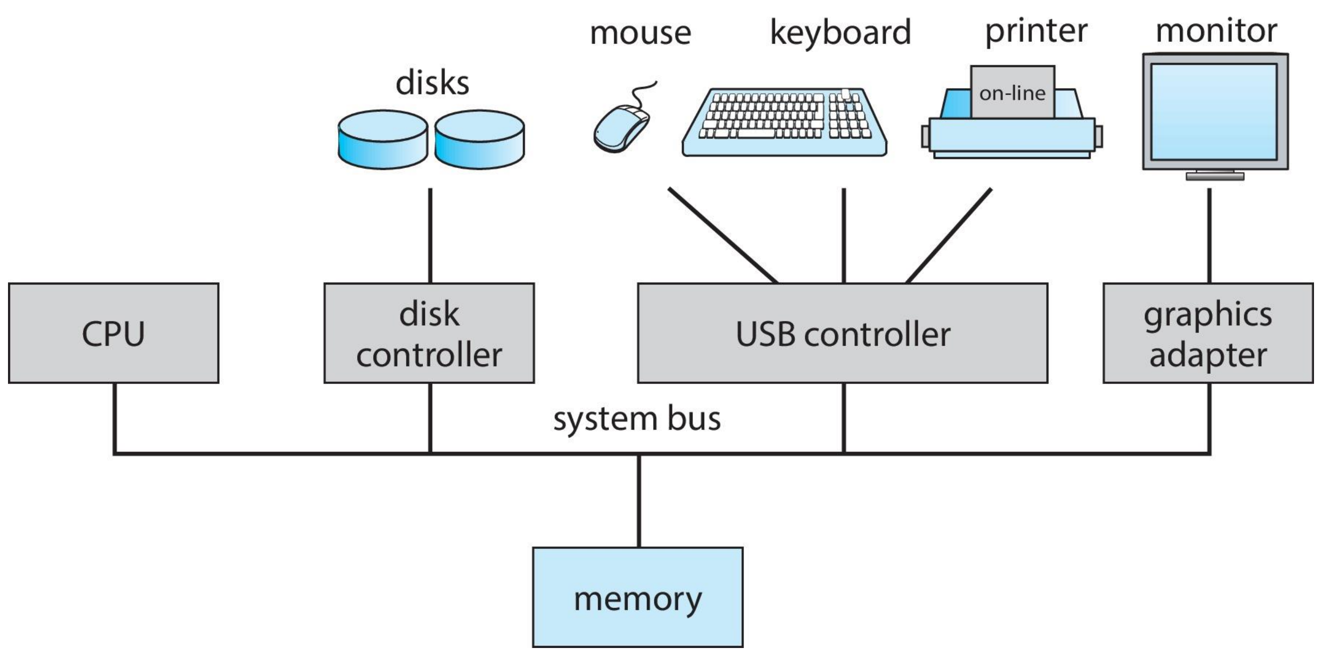
Operating System Definition

- "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
  - 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 additional services to application developers such as databases, multimedia, graphics.
- 1998, Microsoft suit

Overview of Computer System Structure

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

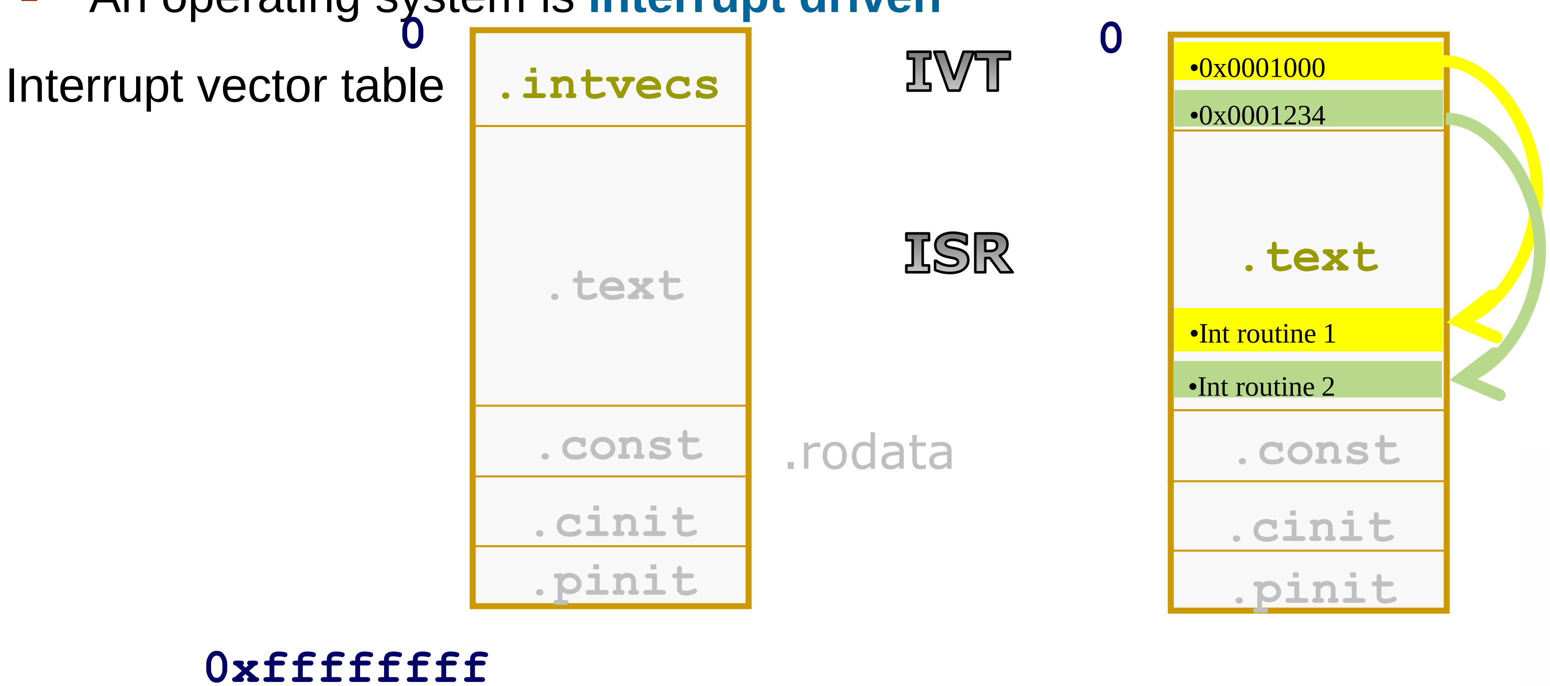


Computer-System Operation

- 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 operation by causing an **interrupt /pooling**

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**

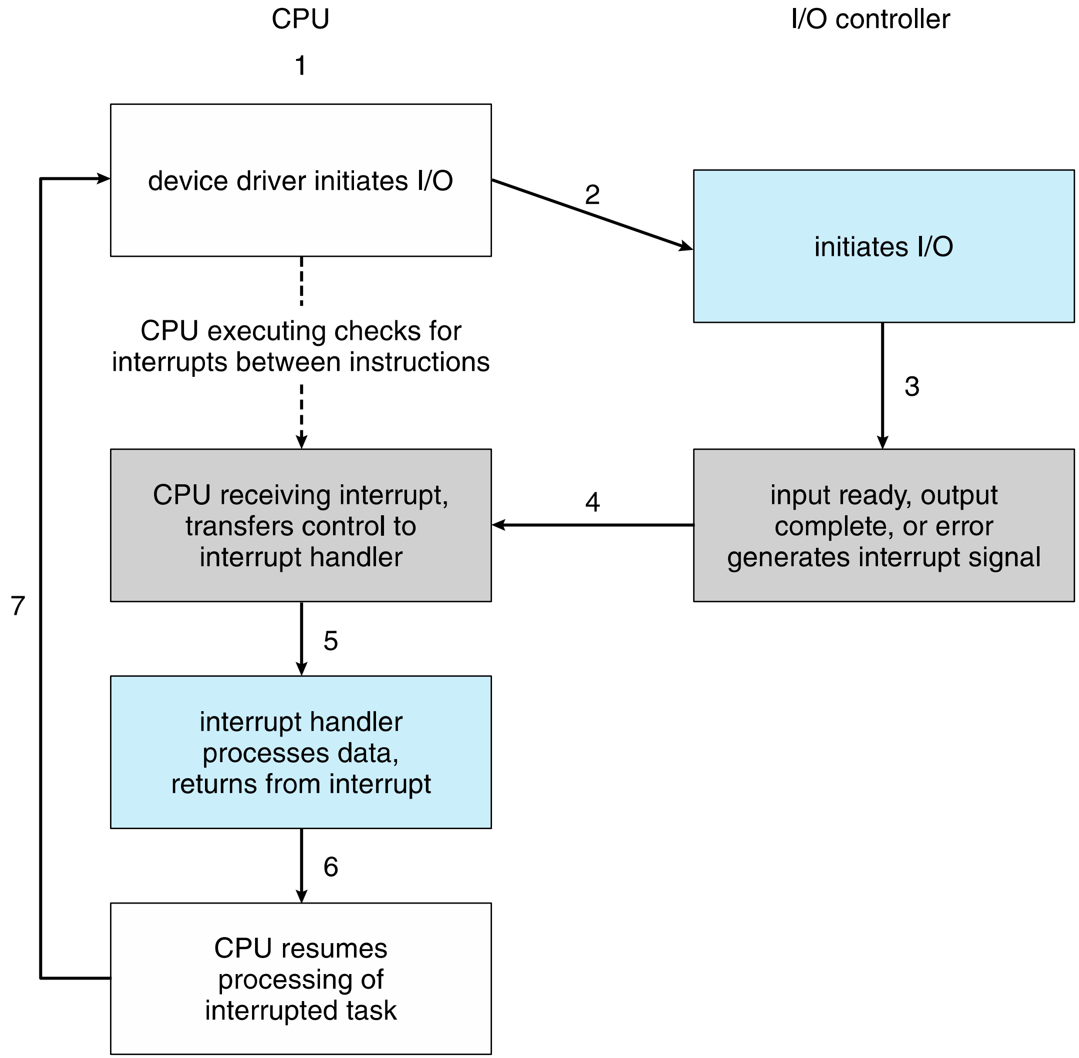


Interrupt Handling

- The operating system preserves the state of the CPU by storing the registers and the program counter
- Determines which type of interrupt has occurred: (Maskable /Non-maskable interrupts)
- Separate segments of code determine what action should be taken for each type of interrupt

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

Interrupt-drive I/O Cycle



I/O Structure (Cont.)

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

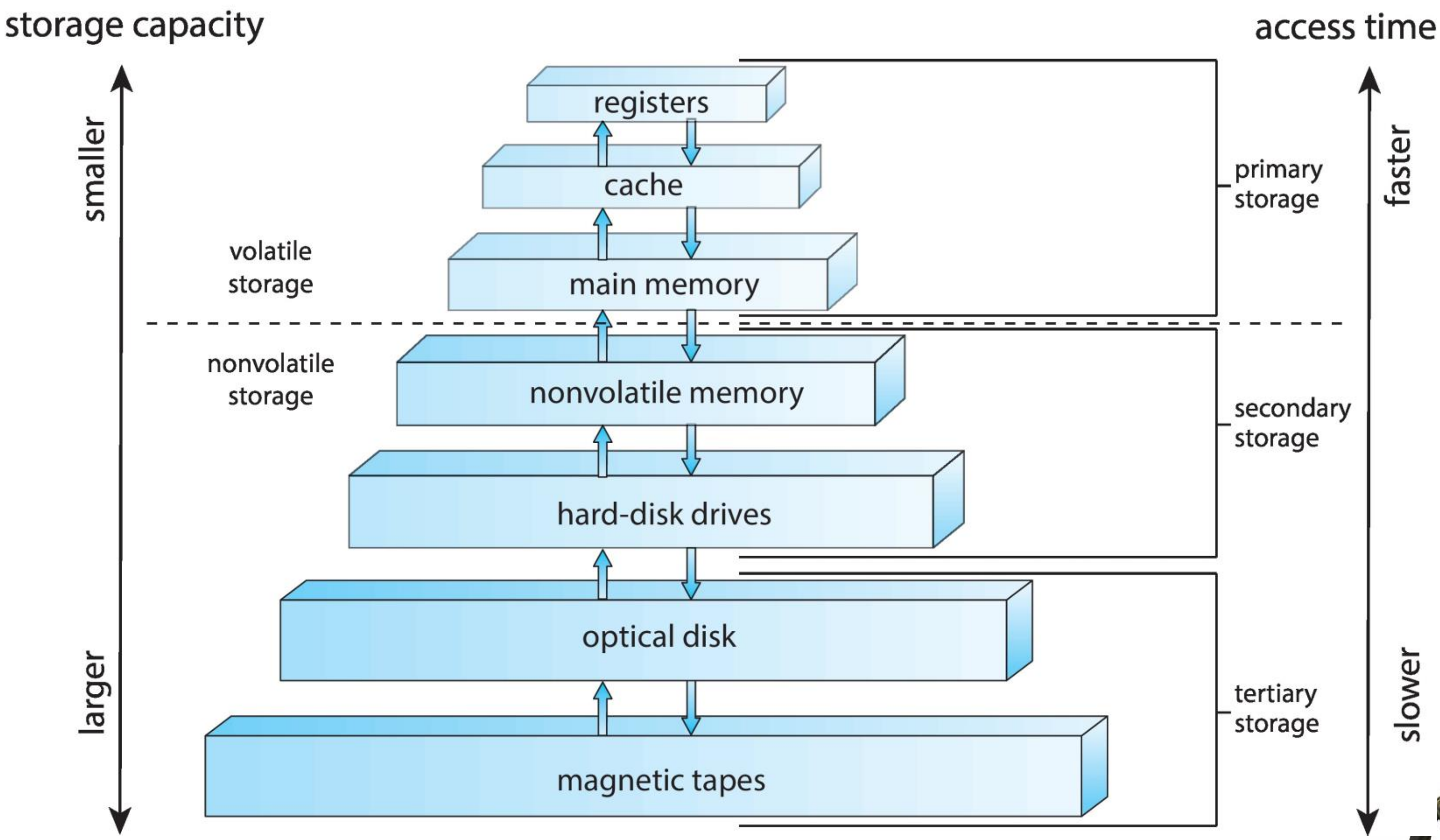
- **System call** – request to the OS to allow user to wait for I/O completion
- **Device-status table** contains entry for each I/O device indicating its type, address, and state
- OS indexes into I/O device table to determine device status and to modify table entry to include interrupt

Storage Structure

Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - **Random access memory (volatile)**, 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**
  - **Non-volatile memory** (**NVM**) devices– faster than hard disks, nonvolatile
    - Various technologies (flash memory).
    - Becoming more popular as capacity and performance increases, price drops

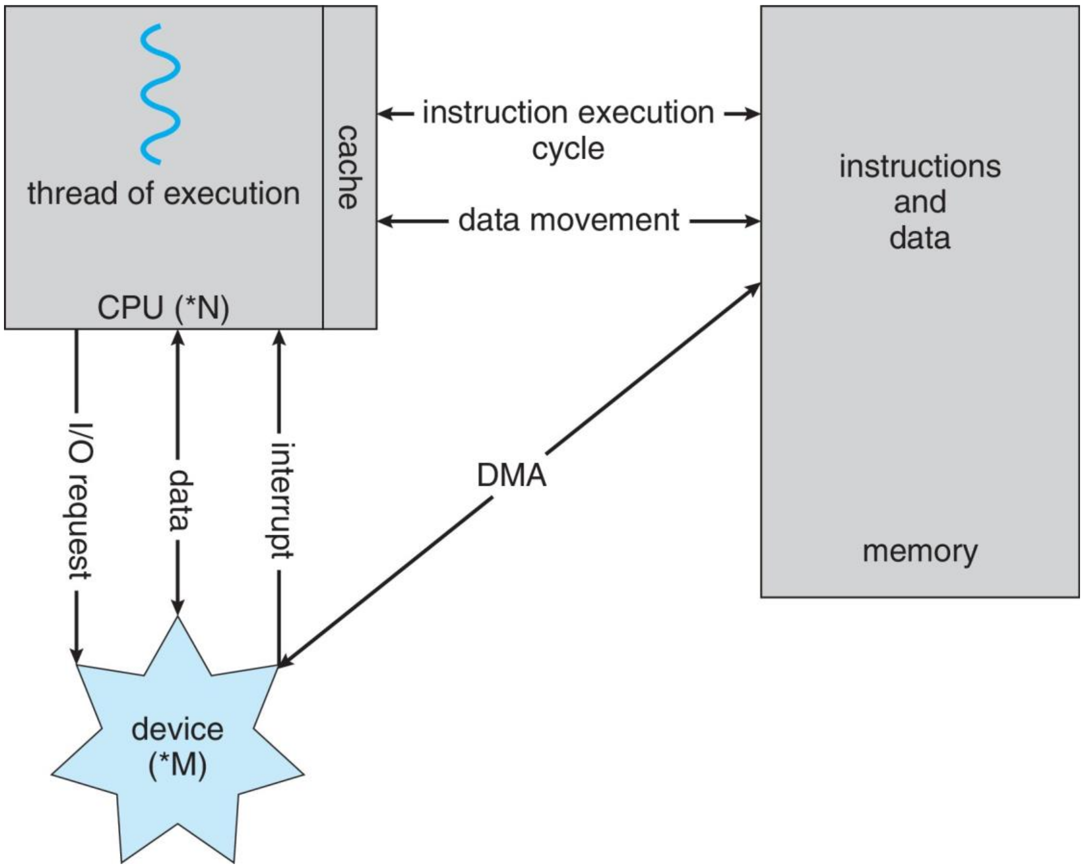
Storage-Device Hierarchy



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

How a Modern Computer Works



A von Neumann architecture

Direct Memory Access Structure

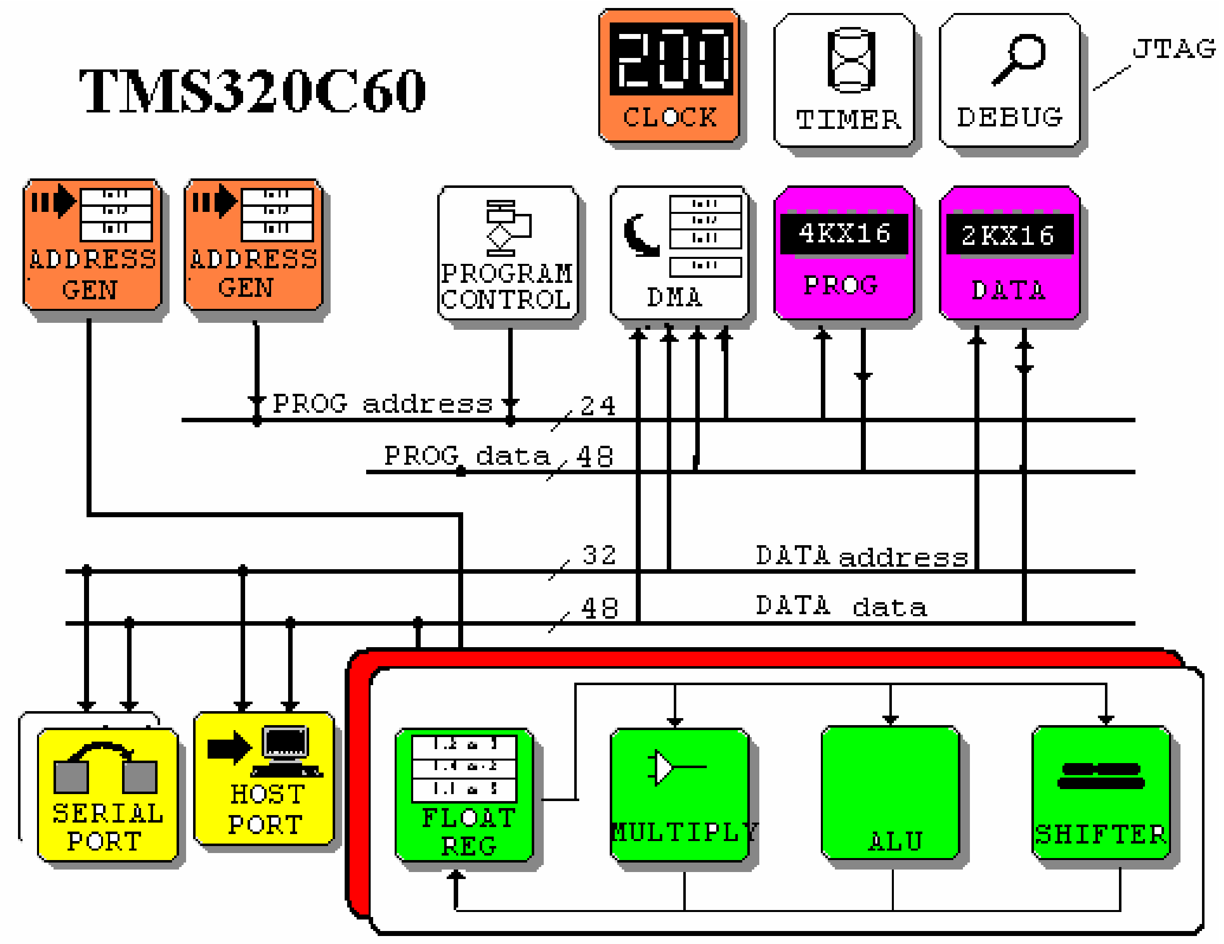
- Used for high-speed I/O devices able to transmit information at close to memory speeds
- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention
- Only one interrupt is generated per block, rather than the one interrupt per byte

Computer System Architecture

Computer-System Architecture

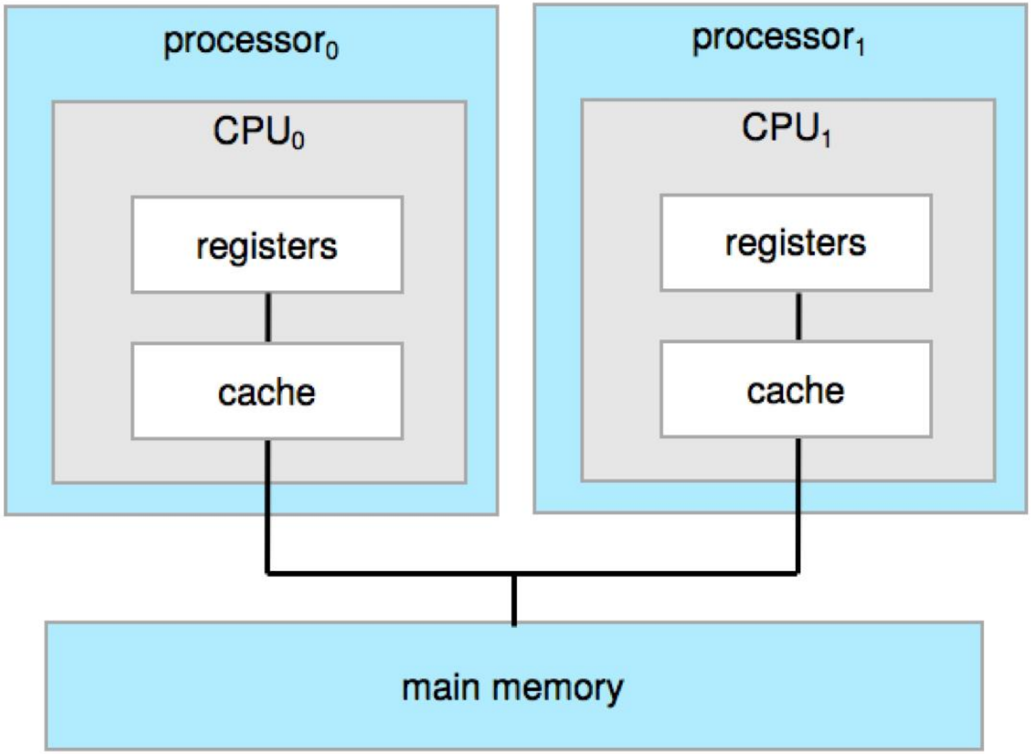
- Most systems use a single general-purpose processor
  - Most systems have special-purpose processors as well
- **Multiprocessors** systems growing in use and importance
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. **Increased throughput**
    2. **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 (SMP)**– each processor performs all tasks

Asymmetric Multiprocessing



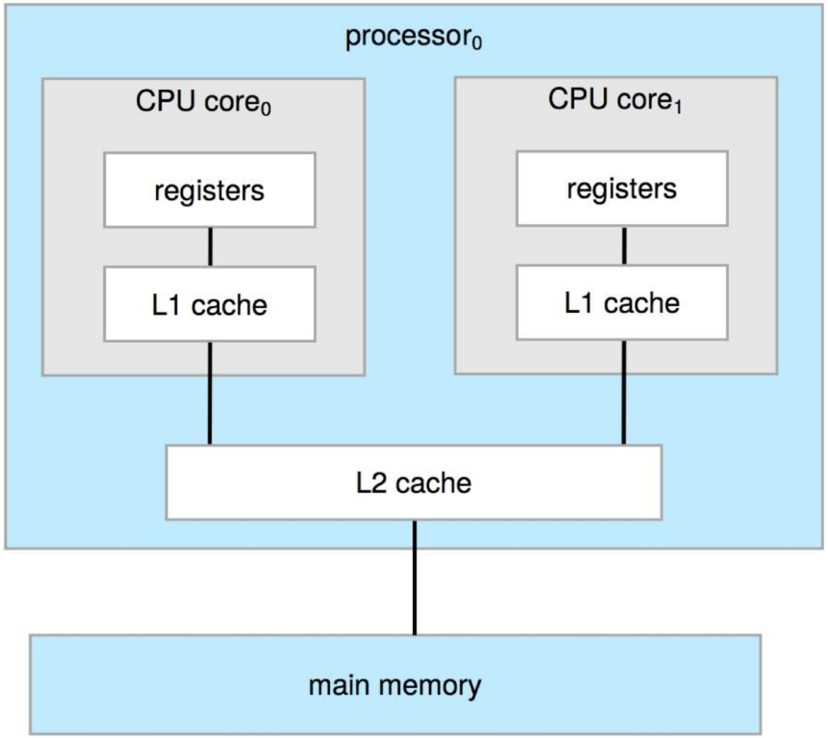
Symmetric Multiprocessing Architecture

Multi-chip and multicore

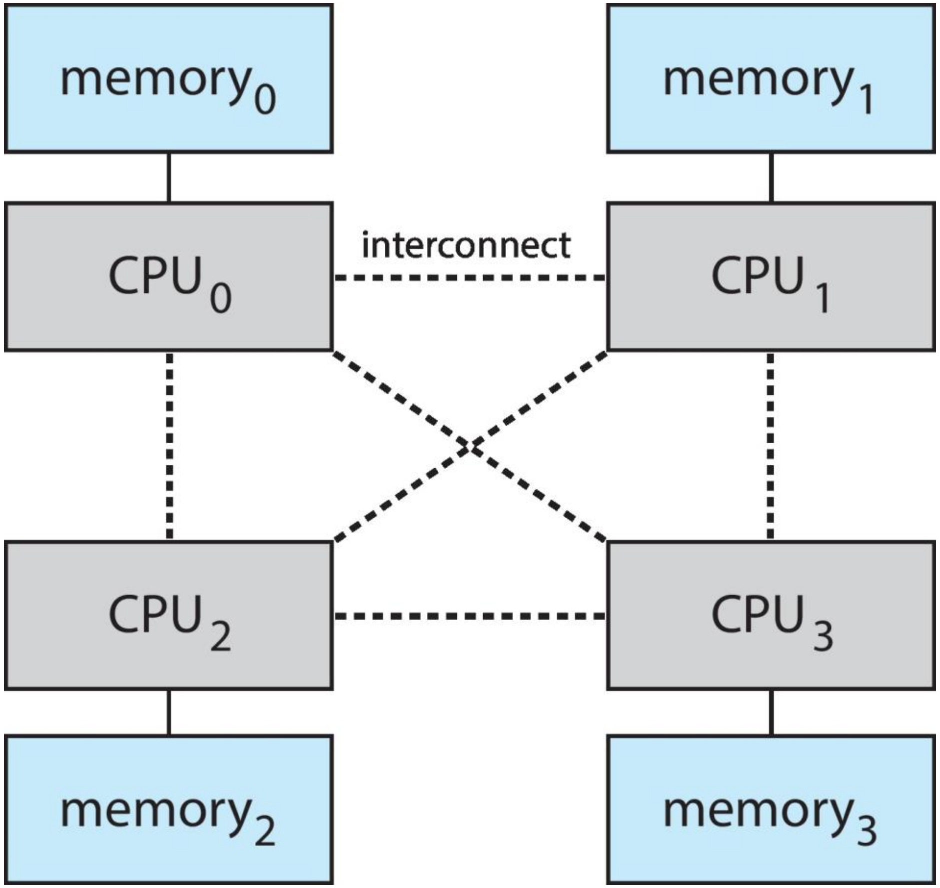


Dual-Core Design

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

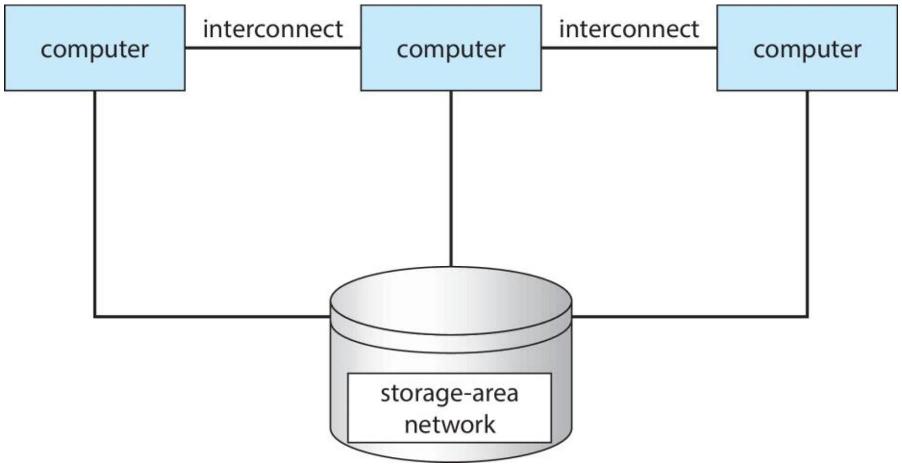


Non-Uniform Memory Access System



Clustered Systems

- Usually sharing storage via a **storage-area network** (**SAN**)



- Provides a **high-availability** service which survives failures
  - The ability to continue providing service proportional to the level of surviving hardware is called **graceful degradation**.
  - The ability to continue, even if there is a failure is exist, **fault tolerant.**
  - **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 (same program)**

Clustered Systems

- Other forms of clusters include **parallel clusters** and clustering over a wide-area network (WAN), Parallel clusters allow multiple hosts to access the same data on shared storage. (Oracle Real Application Cluster, AWS).
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

HADOOP: High Availability Distributed Object Oriented Platform

- A distributed file system that manages data and files across distributed computing nodes.

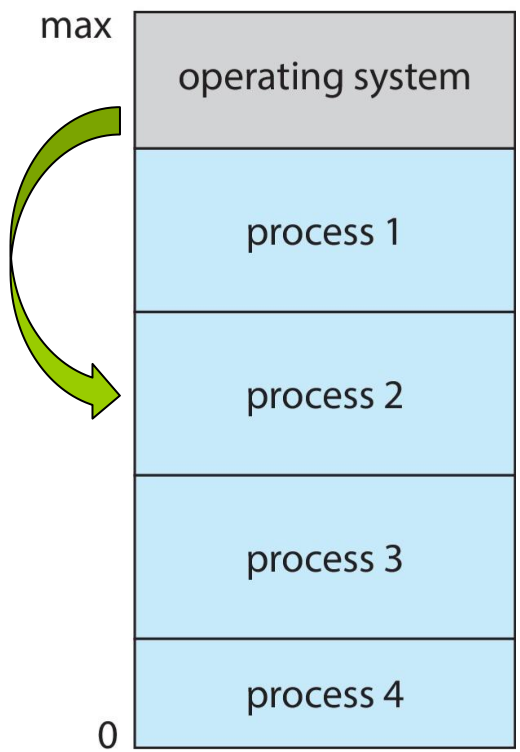
Operating-System Operations

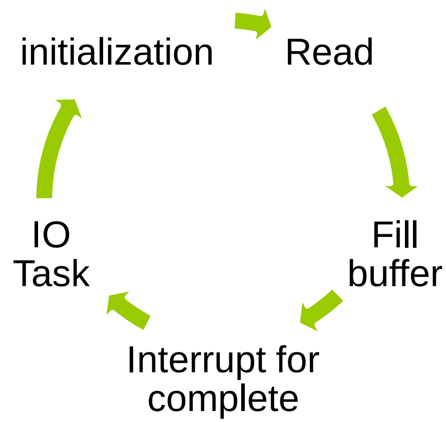
Operating-System Operations

- **Bootstrap program** – simple code to initialize the system, load the kernel
- Kernel loads
- Starts **system daemons** (services provided outside of the kernel)
  - On Linux, the first system program is "systemd".
- 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, invalid memory access)
    - Request for operating system service – **system call**
    - Other process problems include infinite loop, processes modifying each other or the operating system

Multiprogramming (Batch system)

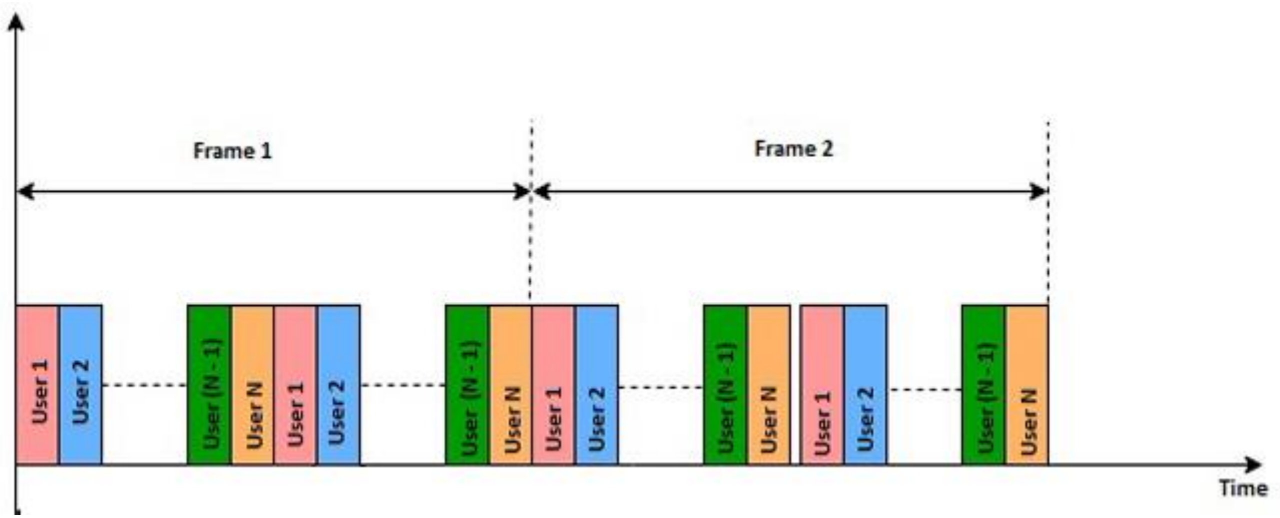
- Single user cannot always keep CPU and I/O devices busy
- Multiprogramming organizes jobs (code and data) so CPU always has *one to execute*
- A subset of total jobs in system is kept in memory (*process*)
- One job selected and run via **job scheduling**
- When job has to wait (for I/O for example),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





Multitasking (Timesharing)

- A logical extension of Batch systems– the CPU switches jobs so *frequently* that users can interact with each job while it is running, creating **interactive** computing
  - Since interactive I/O typically runs at “people speeds
  - **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

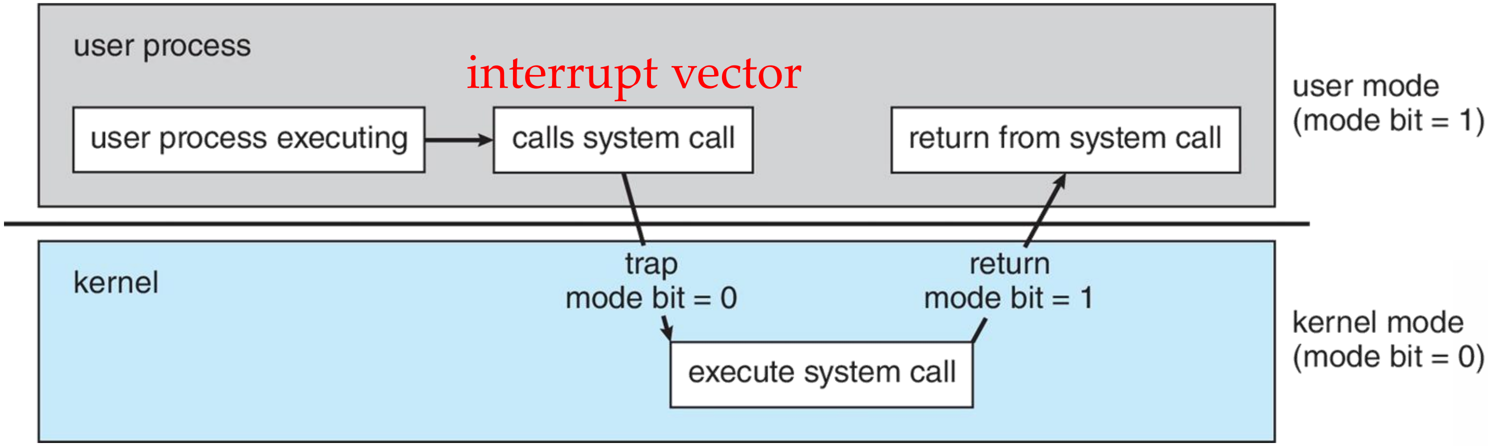


Multitasking (Cont.)

- **Virtual memory** allows execution of processes not completely in memory, memory allows your computer to use part of a permanent storage device (such as a hard disk) as extra memory.

Dual-mode Operation

- **Dual-mode** operation allows OS to protect itself and other system components to execute incorrectly; (incorrect (or malicious) program)
  - **User mode** and **kernel mode**
- **Mode bit** provided by hardware
  - Provides ability to distinguish when system is running user code or kernel code. ⇨ mode bit is “user” = 1⇨ mode bit is “kernel” =0
  - How do we guarantee that user does not explicitly set the mode bit to “kernel”?
- **System call** changes mode to kernel, return from call resets it to user



Dual-mode Operation

- Some instructions designated as **privileged**, only executable in kernel mode. (e.g., I/O control, timer management, and interrupt management).
- Intel processors using 4 protection rings (levels)
- ARMv8 systems, use up to seven modes.

attempt either to :
- Execute an illegal instruction
- Access memory that is not in the user's address space

↓

Hardware interrupts the operating system ,through IVT, ISR

↓

the operating system must terminate the program abnormally.

↓

appropriate error message is given

↓

memory of the program may be dumped (written→ file to examine by programmer)

Timer

- Timer to prevent infinite loop (or 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

File-system Management

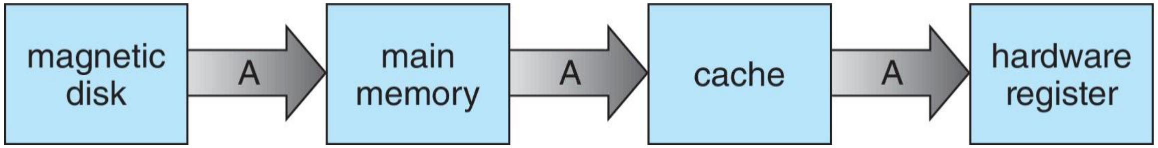
- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file (file extensions)**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what (privileges)
  - 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

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

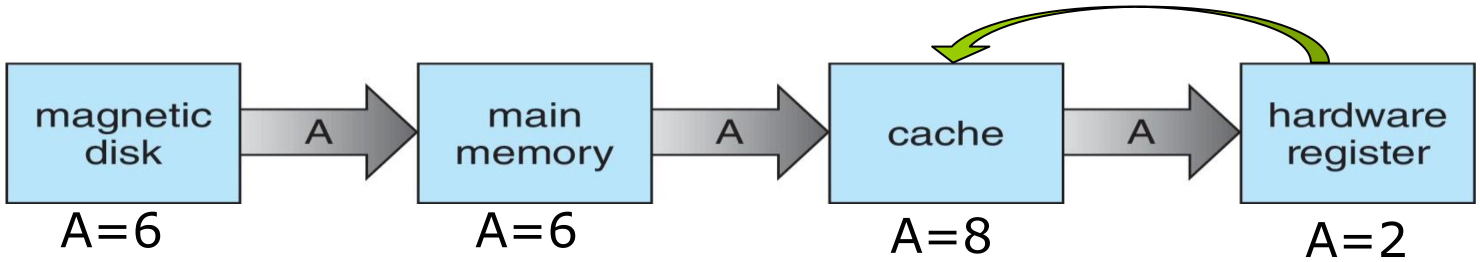
Characteristics of Various Types of Storage

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	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

I/O Subsystem

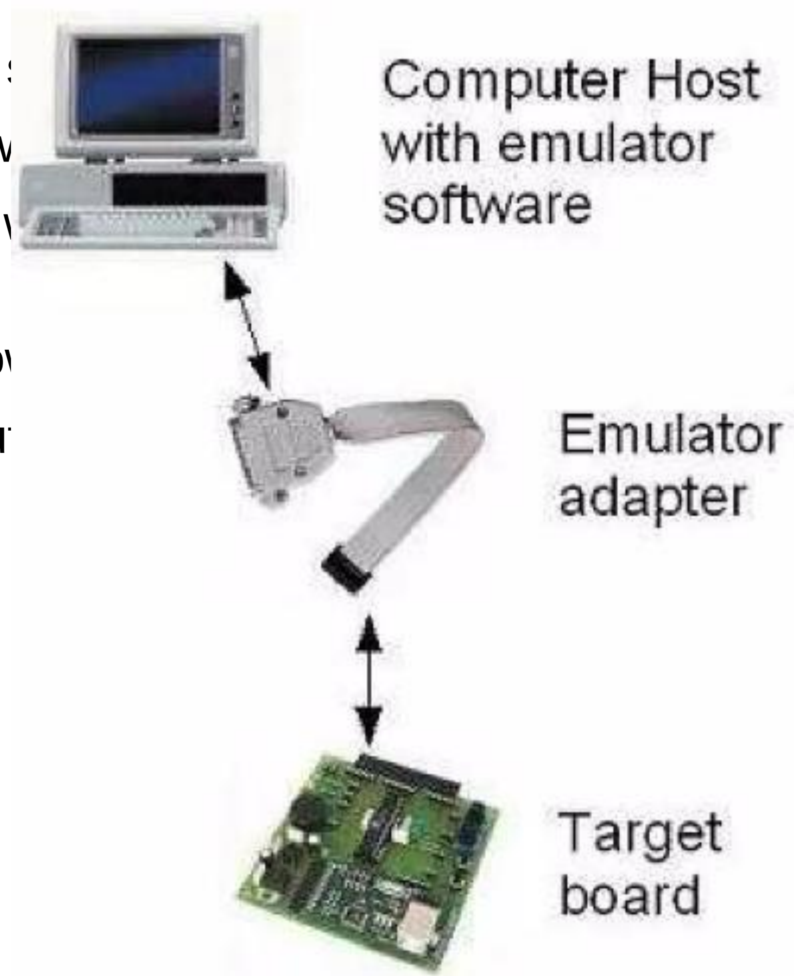
- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (**Simultaneous Peripheral Operations Online**)(the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices

Protection and Security

- **Segmentation faults , infinite loops … etc**
- **Protection** – any mechanism for controlling *access of processes*[1] or users to resources[2] defined by the OS (in a Safe way, be prone to failure and inappropriate access)
- **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 (UID)**, 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

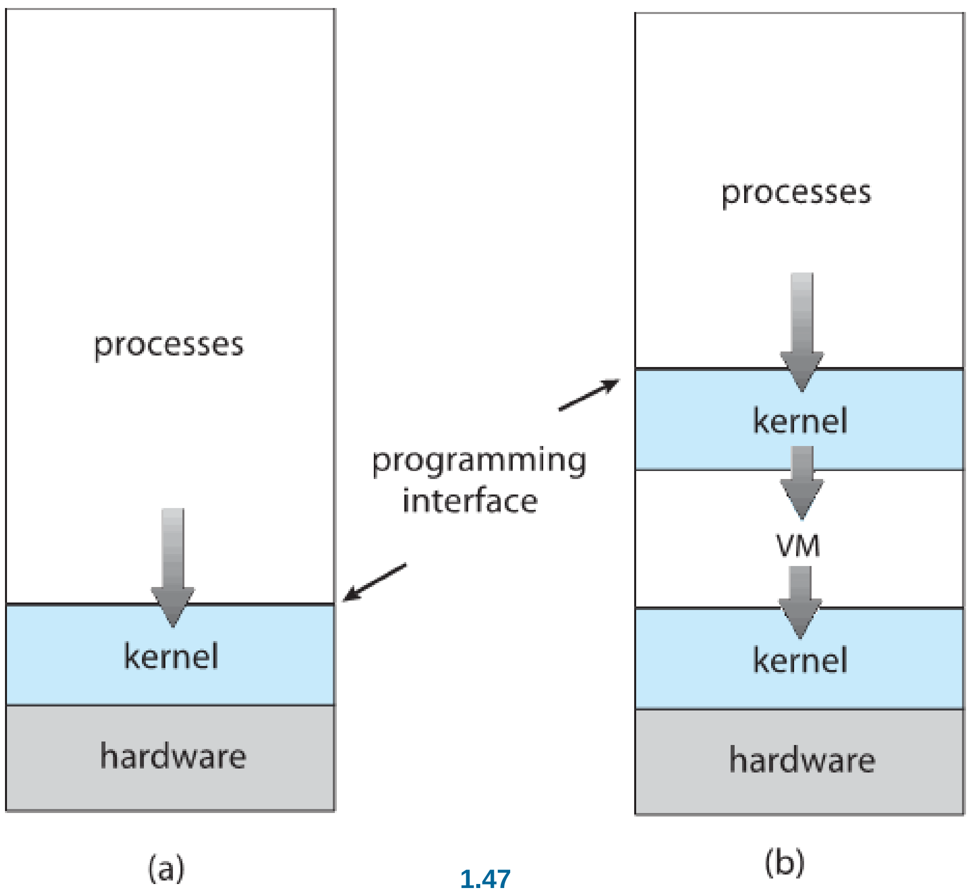
Virtualization

- Allows operating ... er OSes
  - Vast and grov...
- **Emulation** used v... target type (i.e. PowerPC to Intel ...
  - Generally slo...
  - When compu... de – **Interpretation**



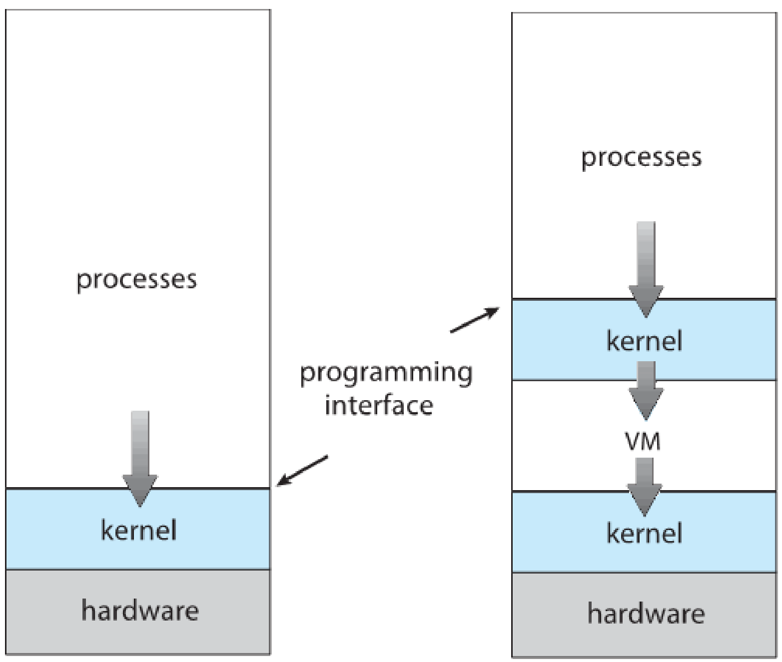
Virtualization (cont.)

- **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



Virtualization (cont.)

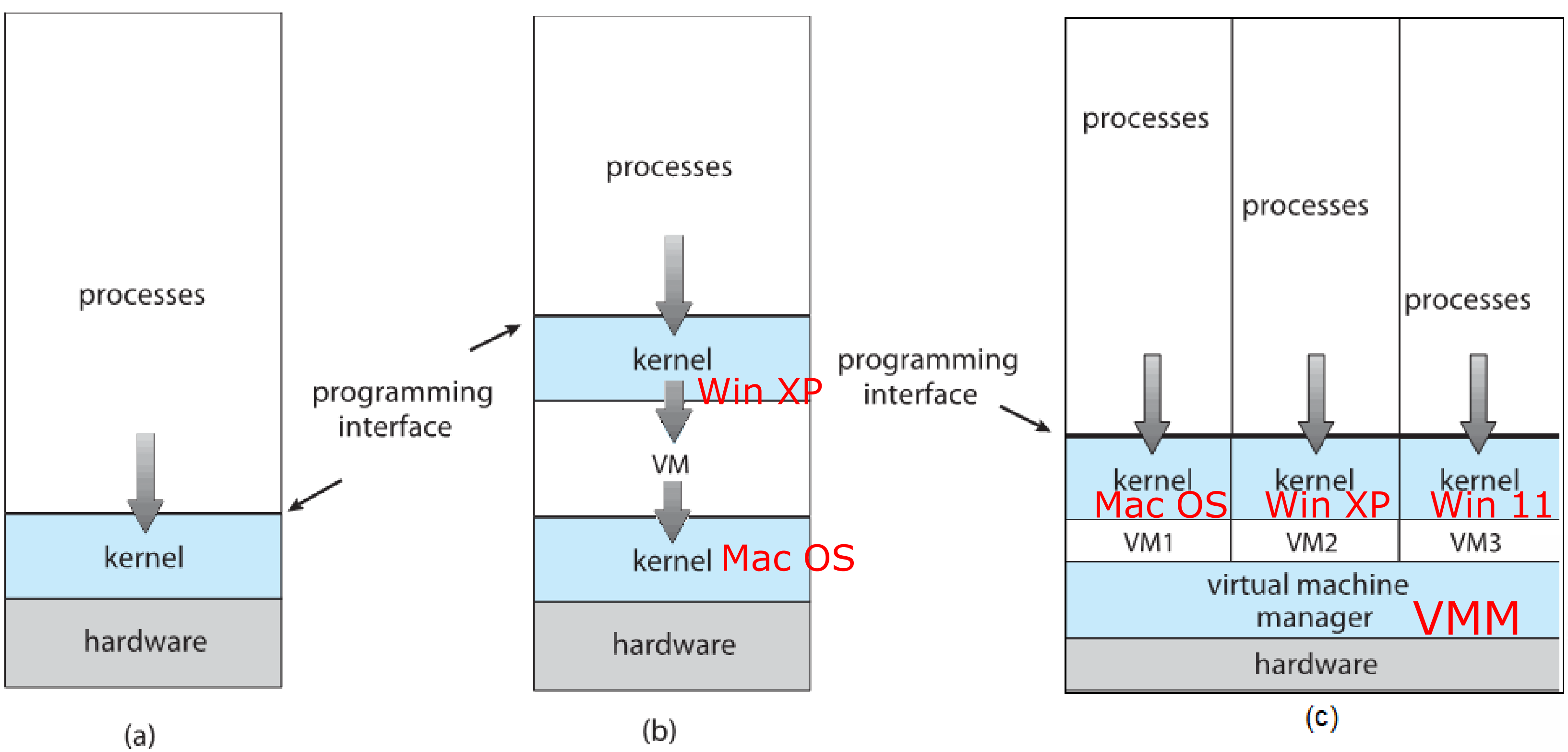
- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
  - Apple laptop running Mac OS X host, Windows as a guest
  - Developing apps for multiple OSes without having multiple systems
  - Quality assurance testing applications without having multiple systems
  - Executing and managing compute environments within data centers



(a) (b)

Computing Environments - Virtualization

VMM (virtual machine manager)can run natively, ESXand Citrix XenServer



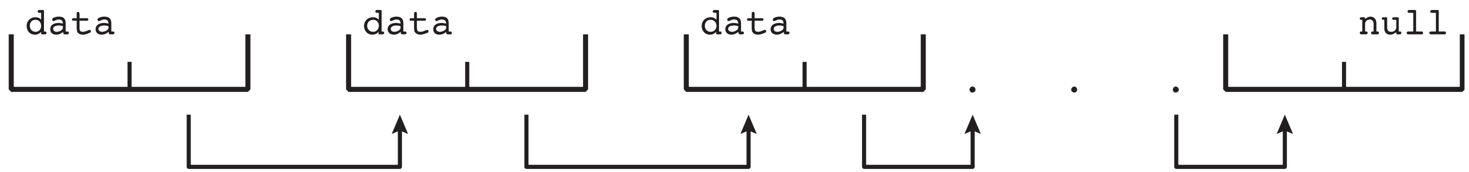
Distributed Systems

- 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

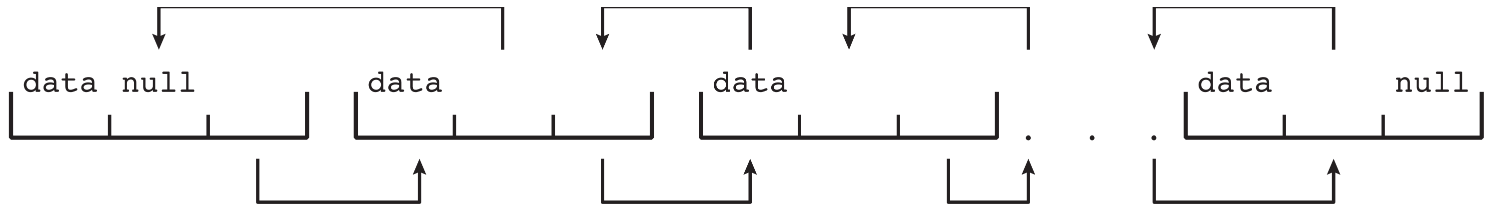
Kernel Data Structure

Kernel Data Structures

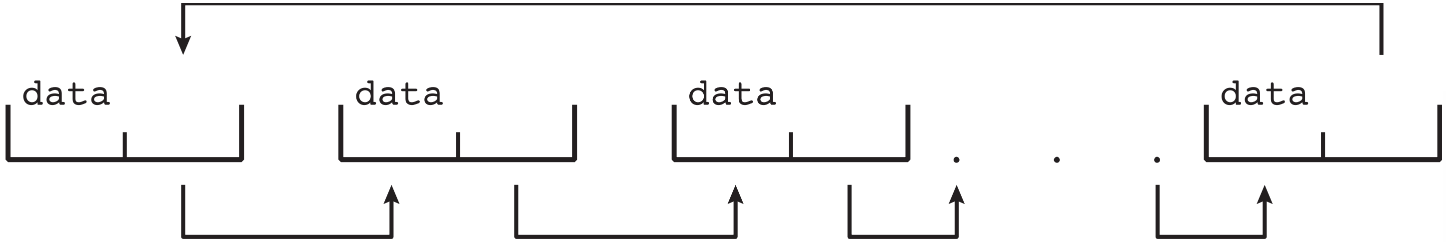
- Many similar to standard programming data structures
- ***Singly linked list***



- ***Doubly linked list***

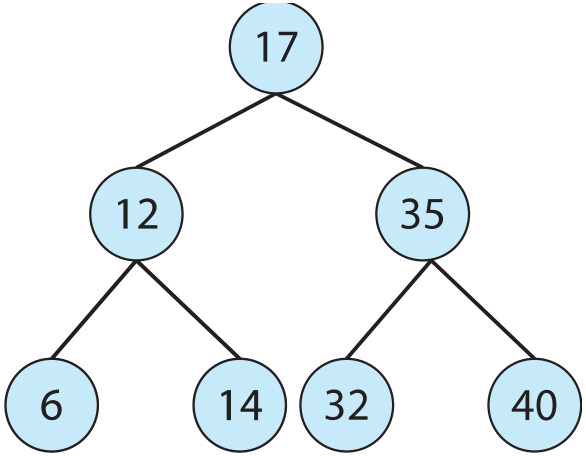


- ***Circular linked list***



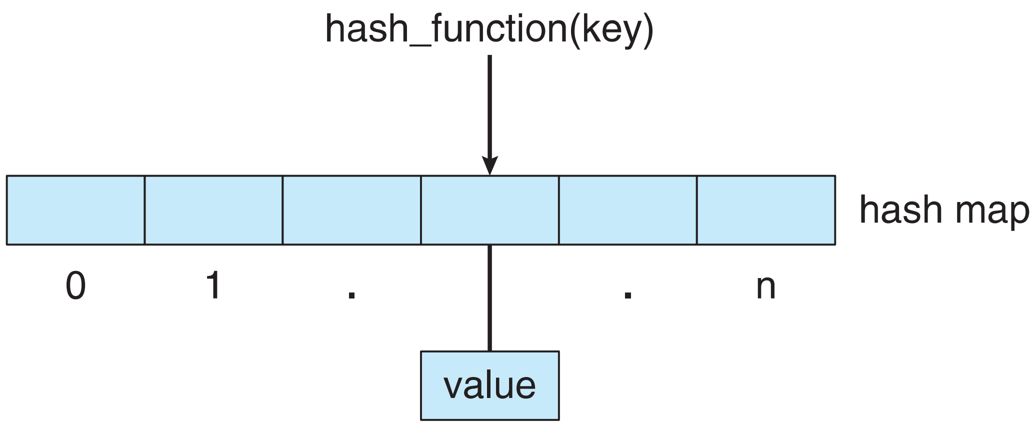
Kernel Data Structures

- **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>`

Computer System Environments

Computing Environments

- Traditional
- Mobile
- Client Server
- Peer-to-Peer
- Cloud computing
- Real-time Embedded

Traditional

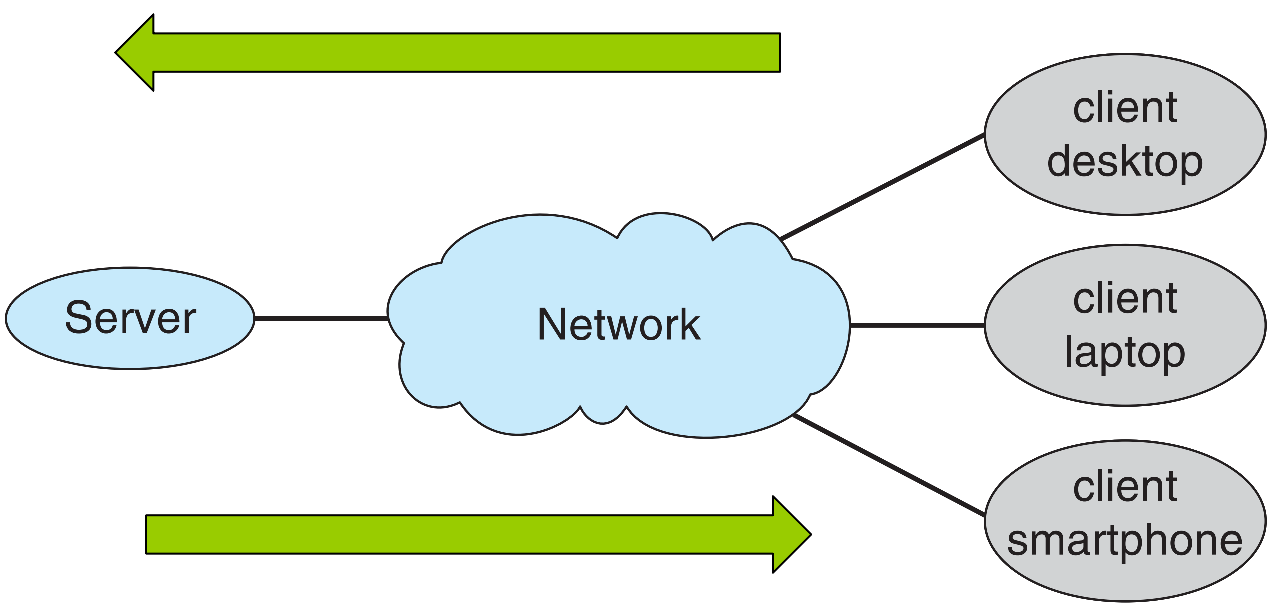
- Stand-alone general-purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers** (**thin clients**) are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking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

Mobile

- Handheld smartphones, tablets, etc.
- What is the functional difference between them and a “traditional” laptop?
- Extra feature – more OS features (GPS, gyroscope ***(6 DOF).***
- 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**

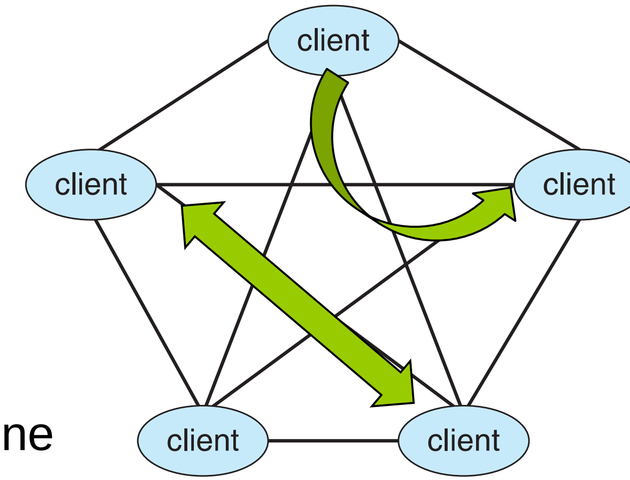
Client Server

- 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*



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
- Determining what services are available:
  - Registers its **service** with *CENTRAL LOOKUP SERVICE* on network, to determine which node provides the service (*Napster)*
  - Another model of distributed system *Broadcast* request for service and respond to requests for service via ***discovery protocol. (****Gnutella)*
- Napster ran into *legal trouble for copyright* infringement, and its services were shut down in 2001

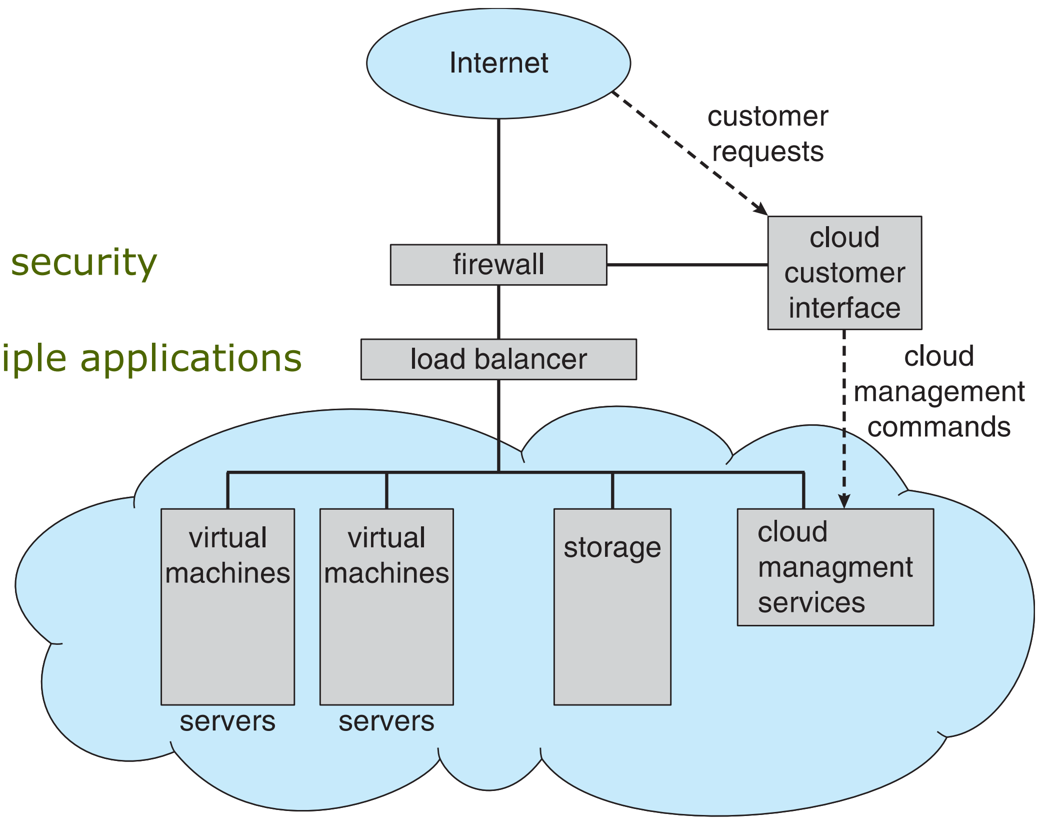


Cloud Computing

- 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 based on usage

requires security

spread traffic across multiple applications



Cloud Computing (Cont.)

- 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)

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 time constraint.
  - Correct operation only if constraints met

End of Chapter 1