Chapter 1: Introduction

Contents

- What Operating Systems Do
- Computer-System Organization
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Computing Environments
- Open-Source Operating Systems

1.1 What Operating Systems Do

- Depends on the point of view
- Users want convenience, **ease of use;** don't care about **resource utilization**
- But shared computer such as **mainframe** or **minicomputer** must keep all users happy
- Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**
- Handheld computers are resource poor, optimized for usability and battery life
- Some computers have little or no user interface, such as embedded computers in devices and automobiles

What Operating Systems Do (Cont'd)

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use

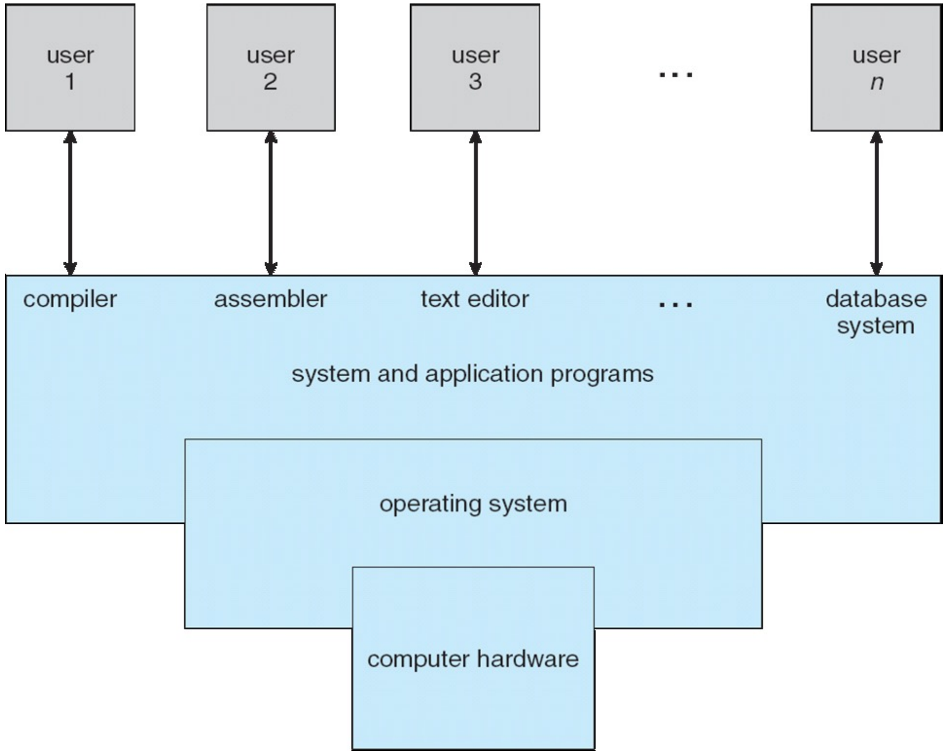
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition

- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is good approximation
- “The one program running at all times on the computer” is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program.
- 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

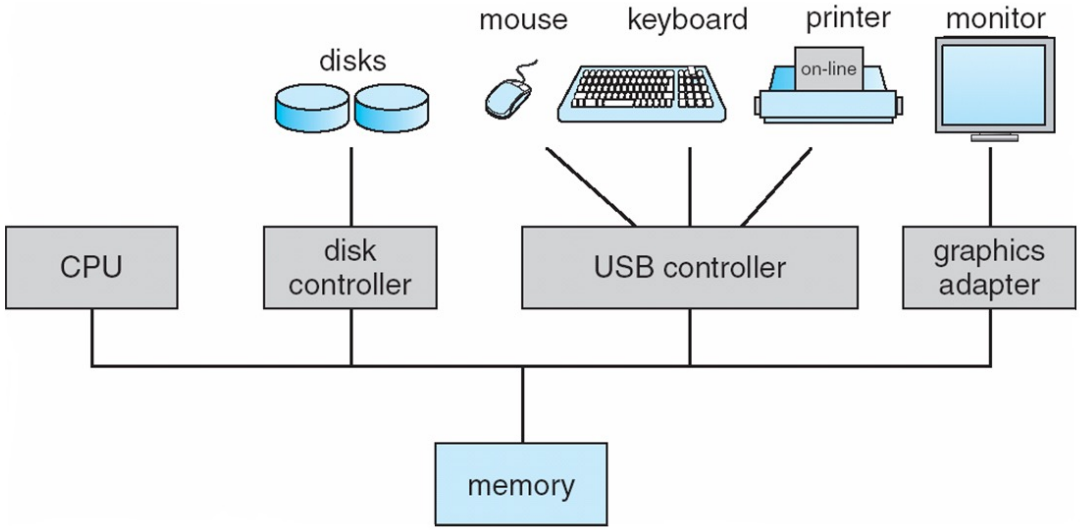
1.2 Computer System Structure

- Computer system can be divi ded into four components:



- Hardware – provides basic computing re sources
  - CPU, memory, I/O devices
- Operating system
  - Controls and coordinates use of hardwar e among various applications and users
- Application programs – define the ways in w hich the system resources are used to solve t he computing problems of the users
  - Word processors, compilers, web br owsers, database systems, video ga mes
- Users - People, machines, other computers

Computer System Organization



- Computer-system operation
  - One or more CPUs, device con trollers connect through com mon bus providing access to s hared memory
  - Concurrent execution of CPUs and devices competing for me mory 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
- CPU moves data from/to main memory to/from regisers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operatio n by causing an interrupt

Common Functions of Interrupts

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request
- An operating system is **interrupt driven**

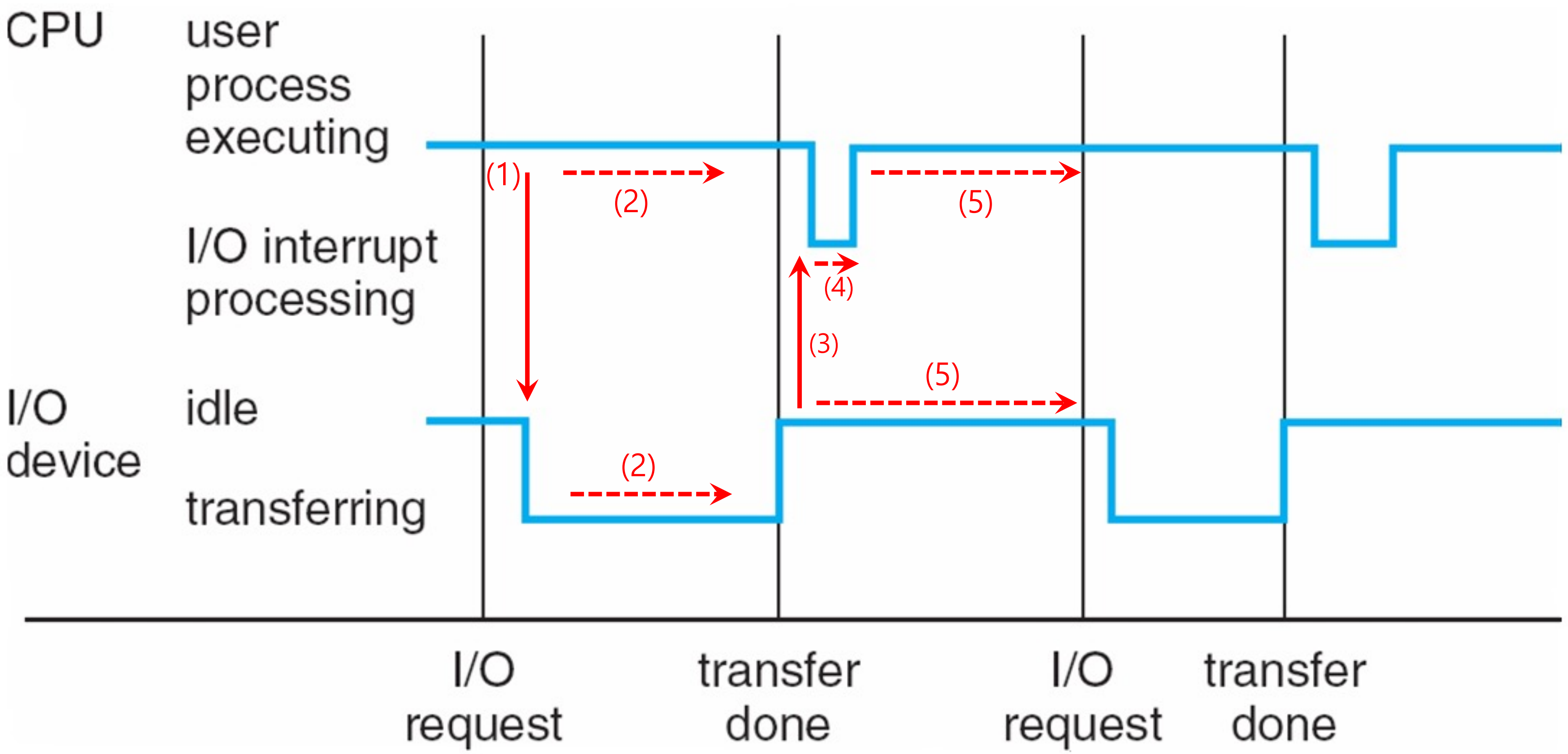
Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter

- Determines which type of interrupt has occurred:
  - **polling**
  - **vectored** interrupt system

- Separate segments of code determine what action should be taken for each type of interrupt

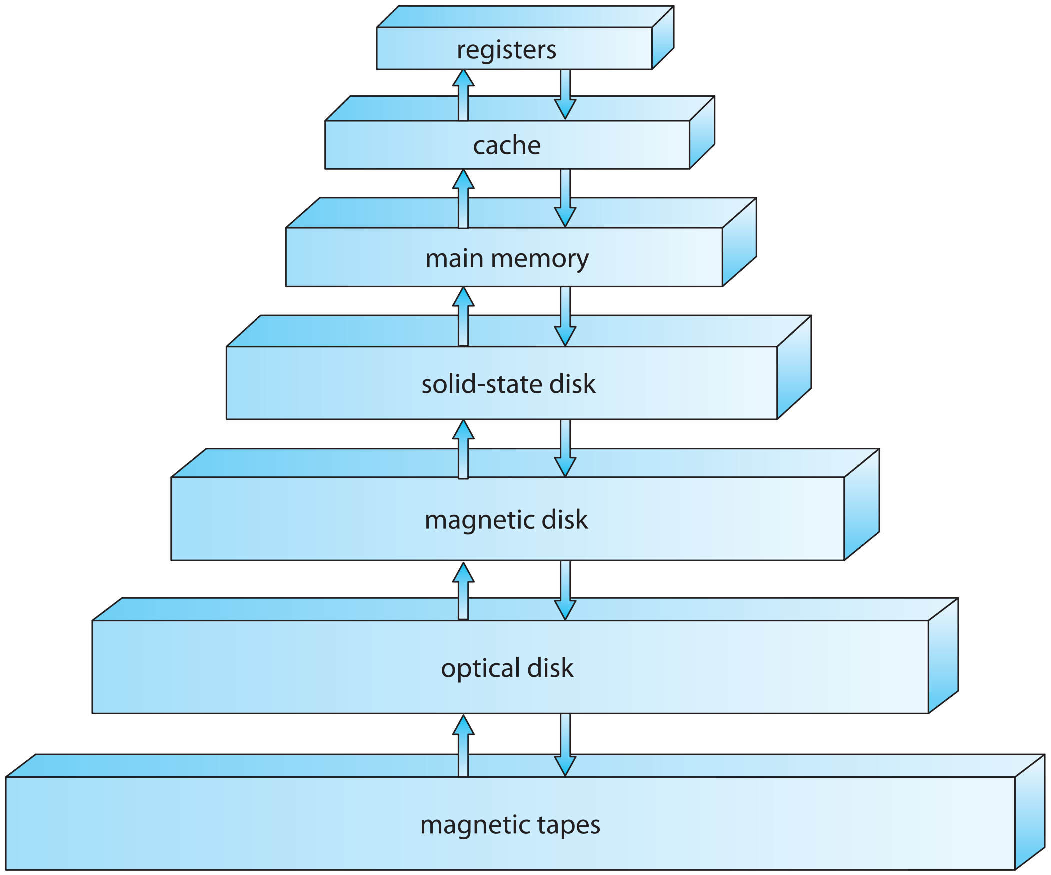
Interrupt Timeline



Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - **Random access**
  - Typically **volatile**
- Secondary storage – extension of main memory that provides large **non volatile** storage capacity
- Magnetic disks – rigid metal or glass platters covered with magnetic rec ording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer
- **Solid-state disks** – faster than magnetic disks, nonvolatile
  - Various technologies
  - Becoming more popular

Storage Hierarchy



- Storage systems organized in hierarc hy
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information into f aster storage system; main memory can be viewed as a cache for second ary storage
- **Device Driver** for each device contro ller to manage I/O
  - Provides uniform interface between co ntroller and kernel

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

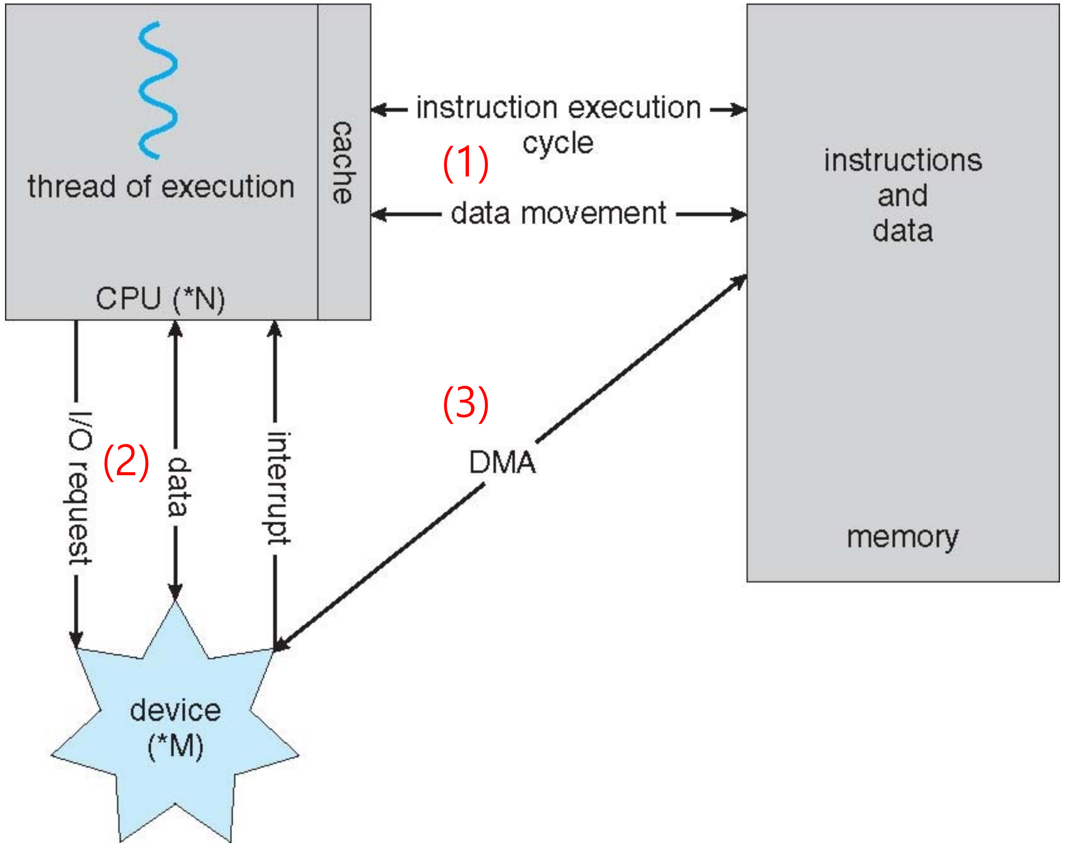
I/O Structure

- After I/O starts, control returns to user program only upon I/O co mpletion
  - 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 pro cessing
- After I/O starts, control returns to user program without waiting fo r 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

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 on e interrupt per byte

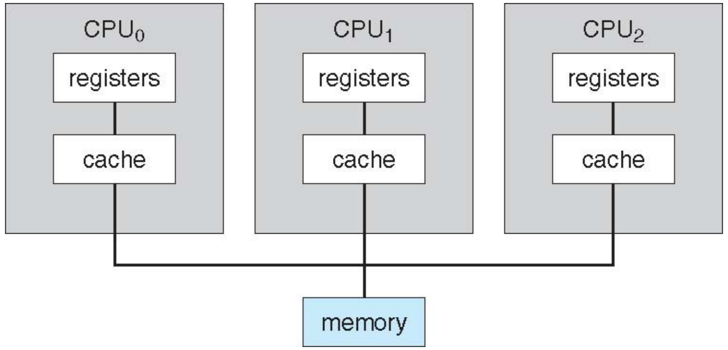
How a Modern Computer Works



A von Neumann architecture

1.3 Computer-System Architecture

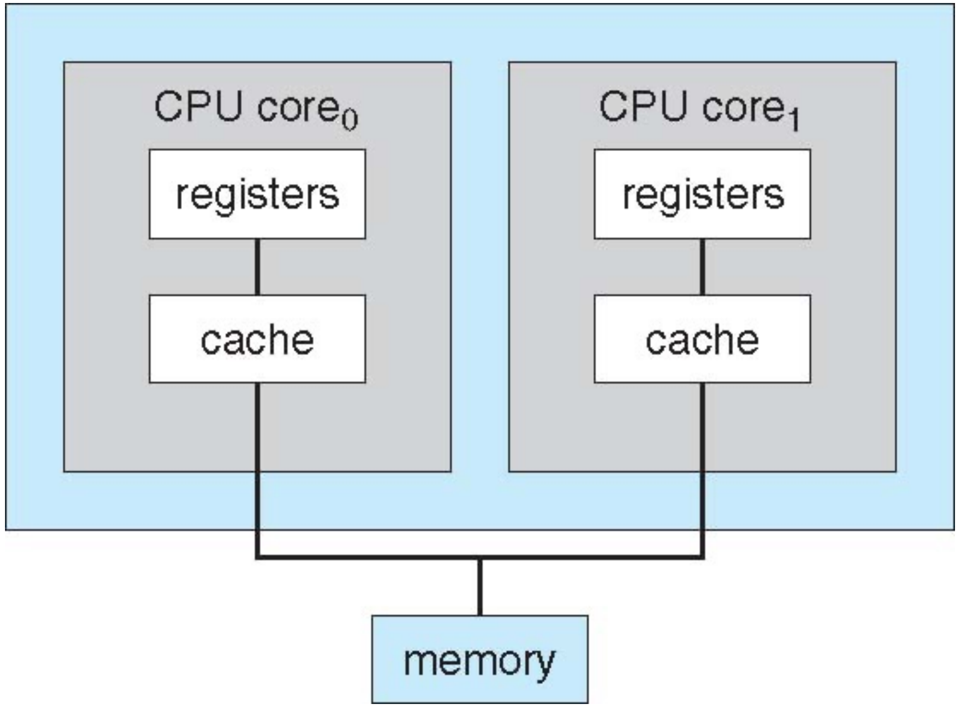
- Most systems use a single general-purpose processor (PDAs through mainframes)
- **Multiprocessors** systems growing in use and importance
  - Also known as **parallel systems**, **multicore system**
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale**
    3. **Increased reliability – graceful degradation** or **fault tolerance**
  - Two types:
    1. **Asymmetric Multiprocessing**
    2. **Symmetric Multiprocessing**
  - **UMA** and **NUMA** variations



Symmetric Multiprocessing Architecture

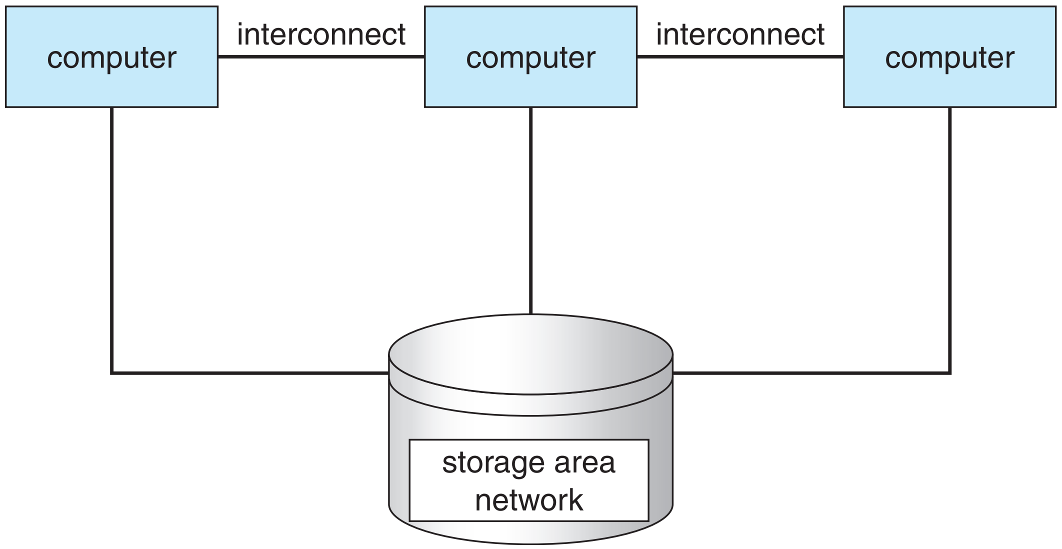
A Dual-Core Design

- Multi-chip and **multicore**



- Systems containing all chips vs. blade servers
  - Chassis containing multiple se parate systems

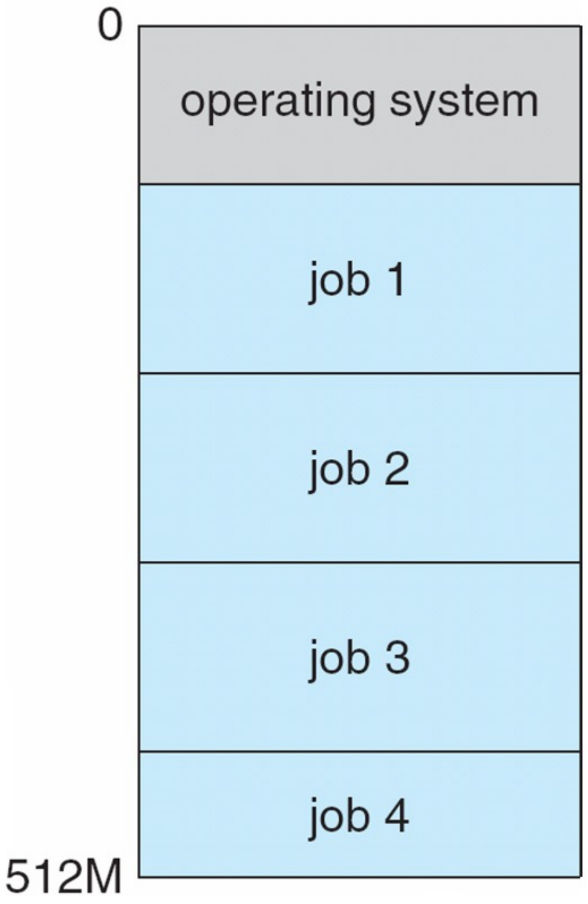
Clustered Systems



- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a **storage-area network (SAN)**
  - Provides a **high-availability** service which survives failures
    - **Asymmetric clustering** vs.
    - **Symmetric clustering**
  - Some clusters are for **high-performance computing (HPC)**
    - Applications must be written to use **parallelization**

1.4 Operating System Structure

- **Multiprogramming** needed f or efficiency
  - A subset of total jobs in syste m is kept in memory
  - Multiprogramming organizes j obs (code and data) so CPU al ways has one to execute. Whe n it has to wait to another job
  - (for I/O for example), OS switc hes
  - job selected and run via **job scheduling**



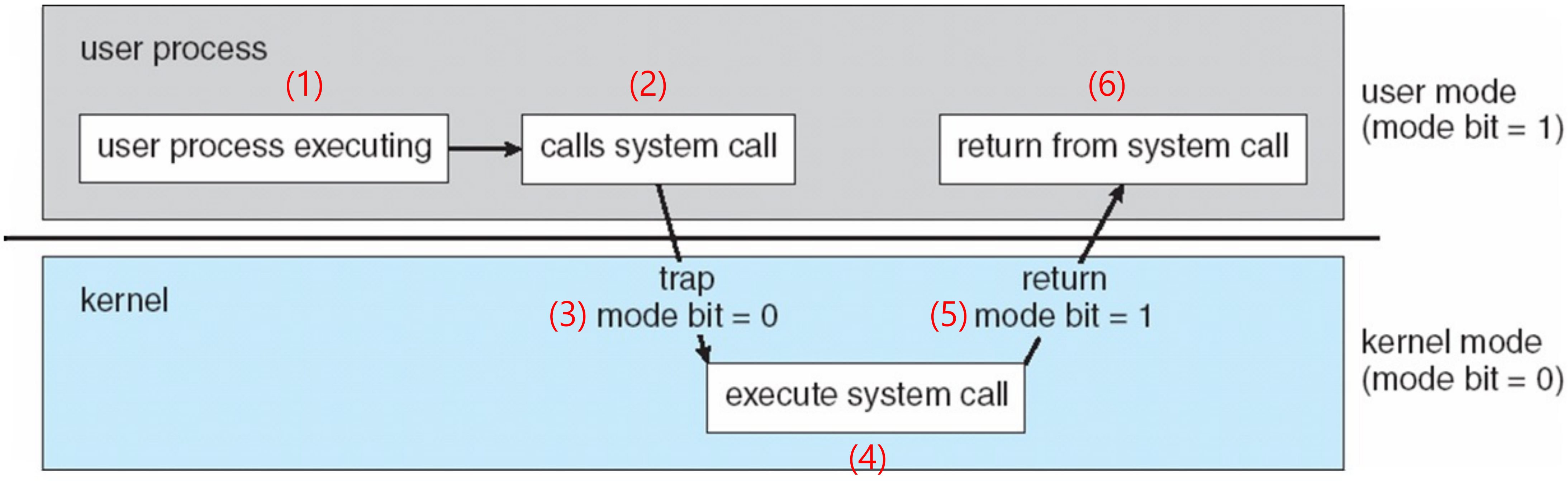
Timesharing

- **Timesharing** (**multitasking**) is logical extension in which CPU switches jobs so frequently that users can interact with each job w hile it is running, creating **interactive** computing
  - **Response time** should be < 1 second
  - Each user has at least one program executing in memory ⇨**process**
  - If several jobs ready to run at the same time ⇨ **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to r un
  - **Virtual memory** allows execution of processes not completely in me mory

1.5 Operating-System Operations

- **Interrupt driven** by hardware
  - Software error or request creates **exception** or **trap** - Division by zero, request for operating system service
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user

Transition from User to Kernel Mode



1.6 Process Management

- A process is a program in execution. It is a unit of work withi n the system. Program is a ***passive entity***, process is an ***activ e entity***.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resource s
- Typically system has many processes, some user, some operat ing system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes

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

1.7 Memory Management

- All data in memory before and after processing
- All instructions in memory in order to execute
- Memory management determines what is in memory 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 w hom
    - Deciding which processes (or parts thereof) and data to move into and out of memory
    - Allocating and deallocating memory space as needed

1.8 Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access metho d (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 dirs
    - 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 man agement is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications

Performance of Various Levels of Storage

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

I/O Subsystem

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

- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temp orarily 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)
  - Drivers for specific hardware devices
  - General device-driver interface

1.9 Protection and Security

- **Protection** – any mechanism for controlling access of processes or user s 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 serv ice
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one p er user
  - User ID then associated with all files, processes of that user to determine access c ontrol
  - Group identifier (**group ID**) allows set of users to be defined and controls manage d, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

1.11 Computing Environments - Traditio nal

- 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 **fi rewalls** to protect home computers from Internet attacks

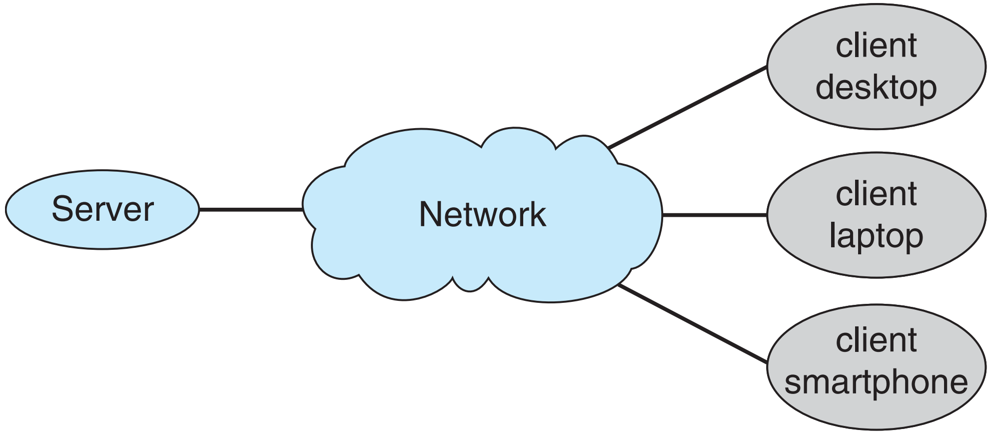
Computing Environments - Mobile

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

Computing Environments – Distributed

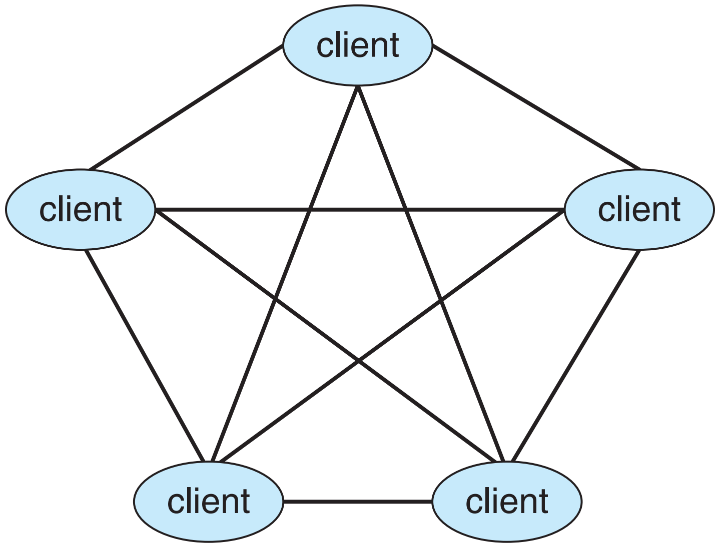
- Collection of separate, possibly heterogeneous, systems netw orked together
- **Network** is a communications path, **TCP/IP** most common
  - **Local Area Network** (**LAN**)
  - **Wide Area Network** (**WAN**)
  - **Metropolitan Area Network** (**MAN**)
  - **Personal Area Network** (**PAN**)
- **Network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

Computing Environments – Client-Server



- Dumb terminals supplanted by smart PCs
- Many systems now servers, re sponding to requests generat ed by clients
  - Compute-server system provid es an interface to client to req uest services (i.e., database)
  - File-server system provides int erface for clients to store and retrieve files

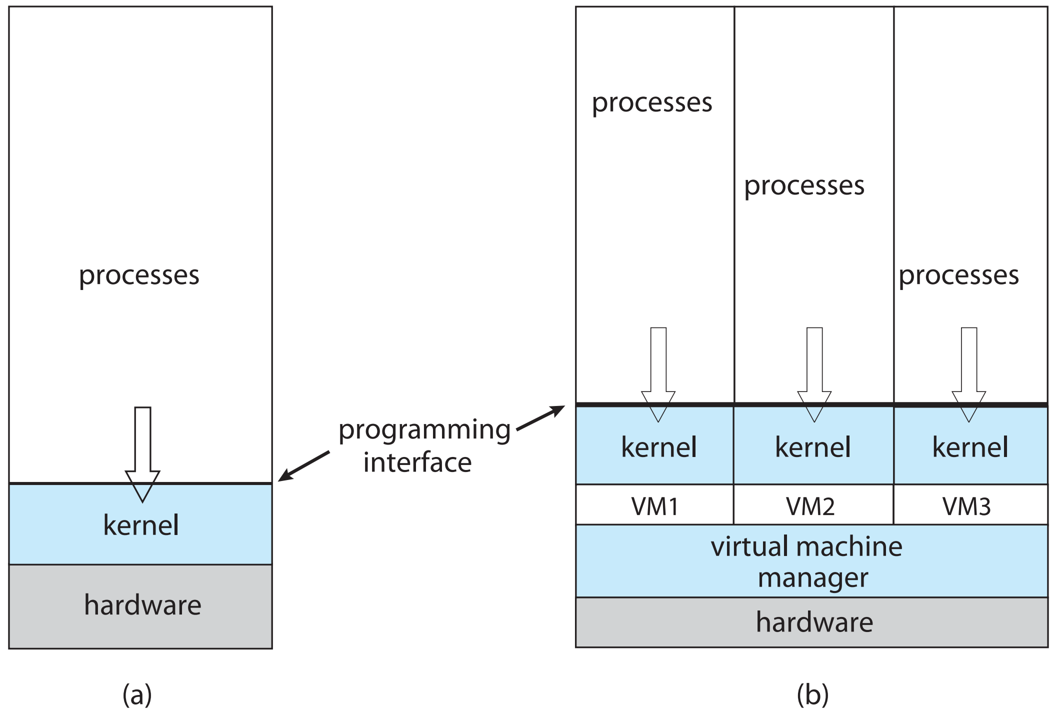
Computing Environments - Peer-to-Peer



- 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 ***discover y 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
- **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** provides virtualization services

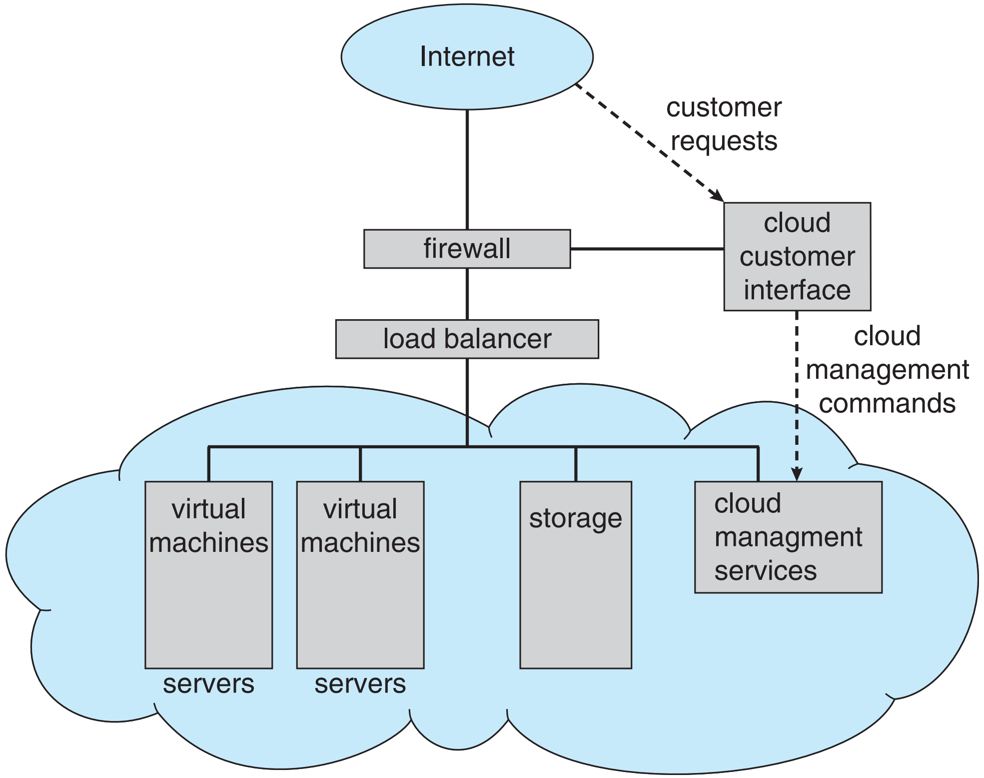


Computing Environments – Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization as based on virtualization
  - Amazon **EC2** has thousands of servers, millions of VMs, PBs of storage available a cross 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 Inter net (i.e a database server)
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e. st orage available for backup use)

Computing Environments – Cloud Computing

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



Computing Environments – Real-Time Embedded Systems

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