Chapter 1: OS Introduction

Content

- Topics
  - Overview of what Operating Systems do
  - Computer System Organization and Architecture
  - Operating System Structure and Operations
  - Process Management
  - Memory Management
  - Storage Management
  - Protection and Security
  - Distributed Systems
  - Special Purpose Systems
- Objectives
  - To provide a grand tour of the major operating systems components
  - To provide coverage of basic computer system organization

Operating System Definition (1)

- A program that acts as an intermediary between a user, and its application programs, of a computer and the computer hardware
- Operating System (OS) 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
- OS is a resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use

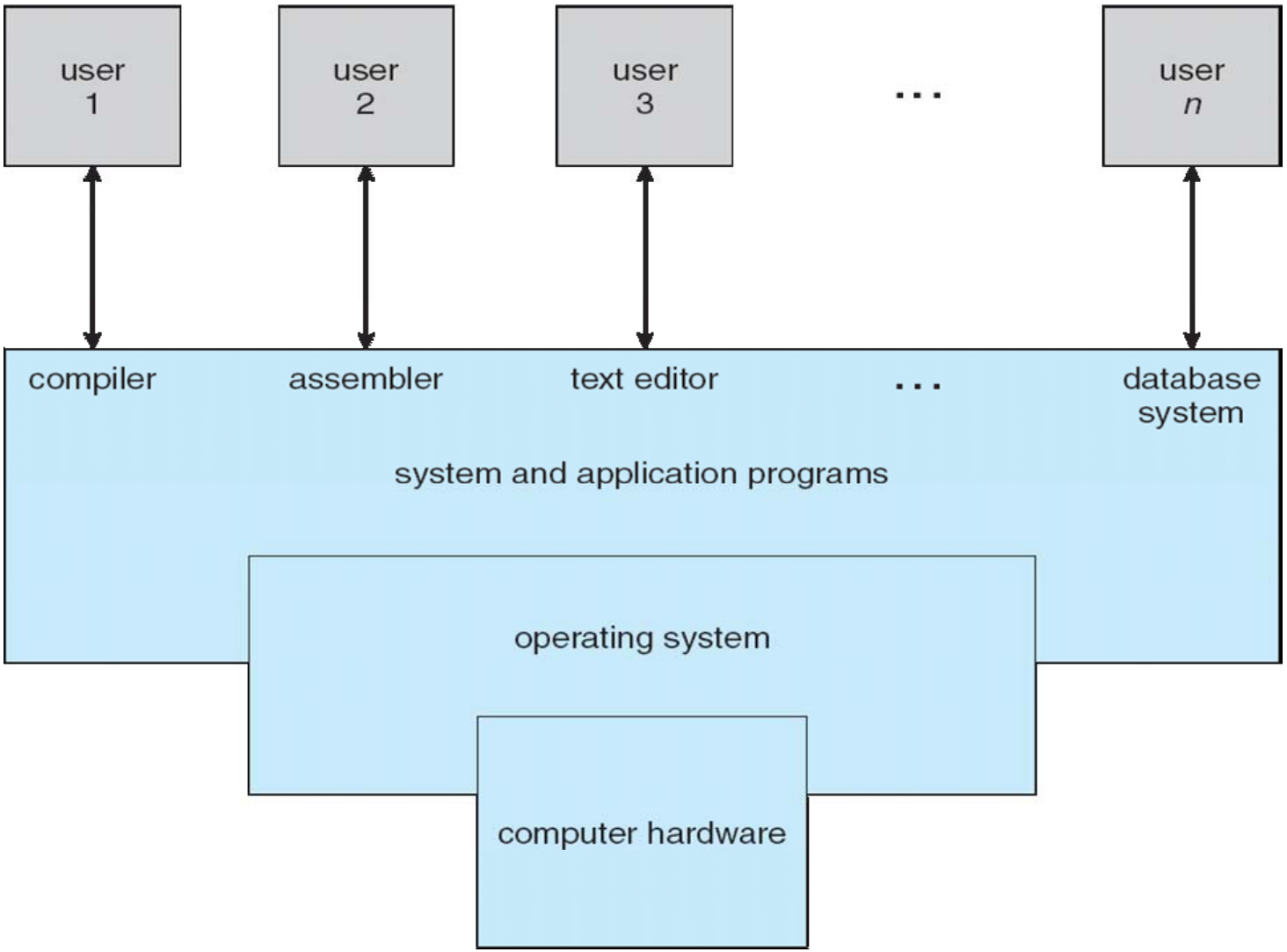
Operating System Definition (2)

- OS is a control program
  - Controls execution of programs to prevent errors and improper use of the computer
- "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
- "bootstrap program" is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as firmware
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Computer System Structure

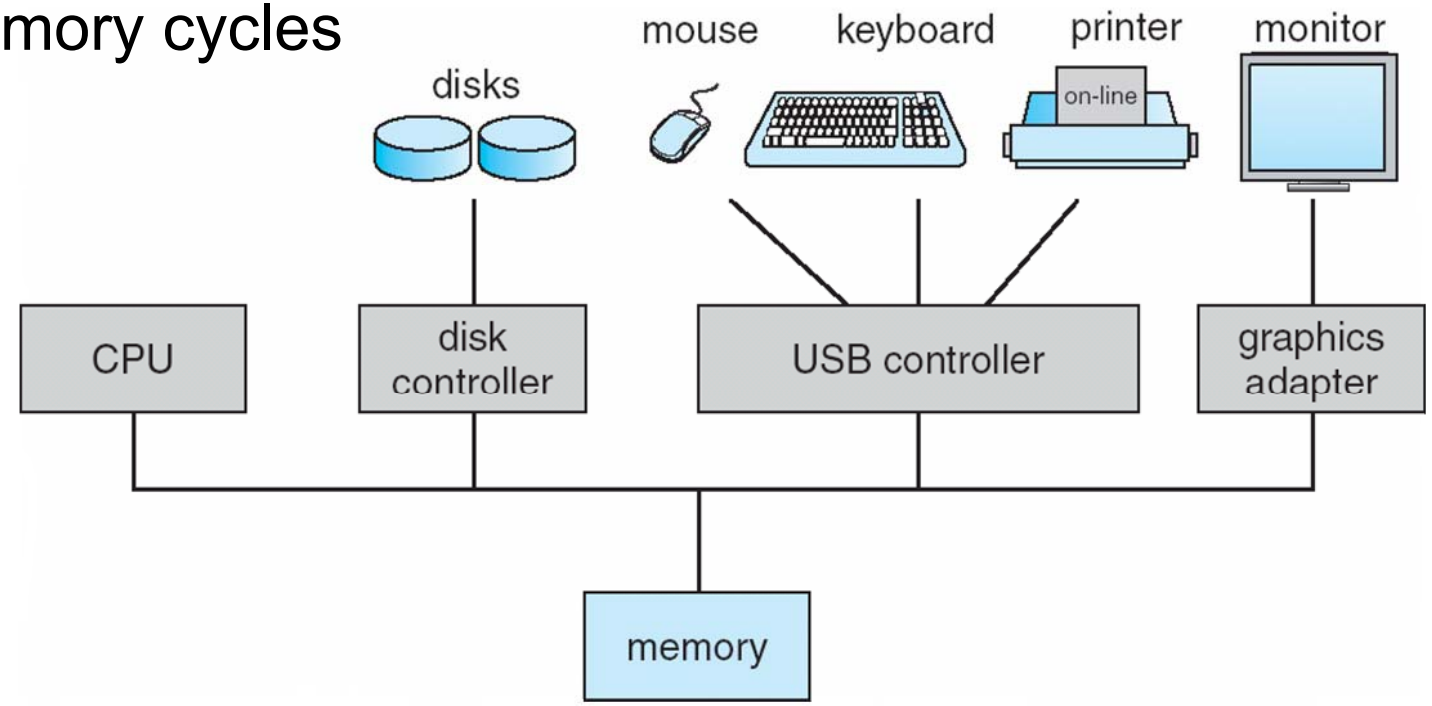
- A computer system can be divided into four main components
  - Hardware – provides basic computing resources
    - CPU (Central Processing Unit), memory, I/O (in/out) devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which 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

Four Components of a Computer System



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
  - *E.g.*, disks, network interface, GPUs (Graphical PU)
- Each device controller has a local memory buffer
  - 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

I/O Structure – Two Alternatives

- 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 operating system 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
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt

Direct Memory Access Structure

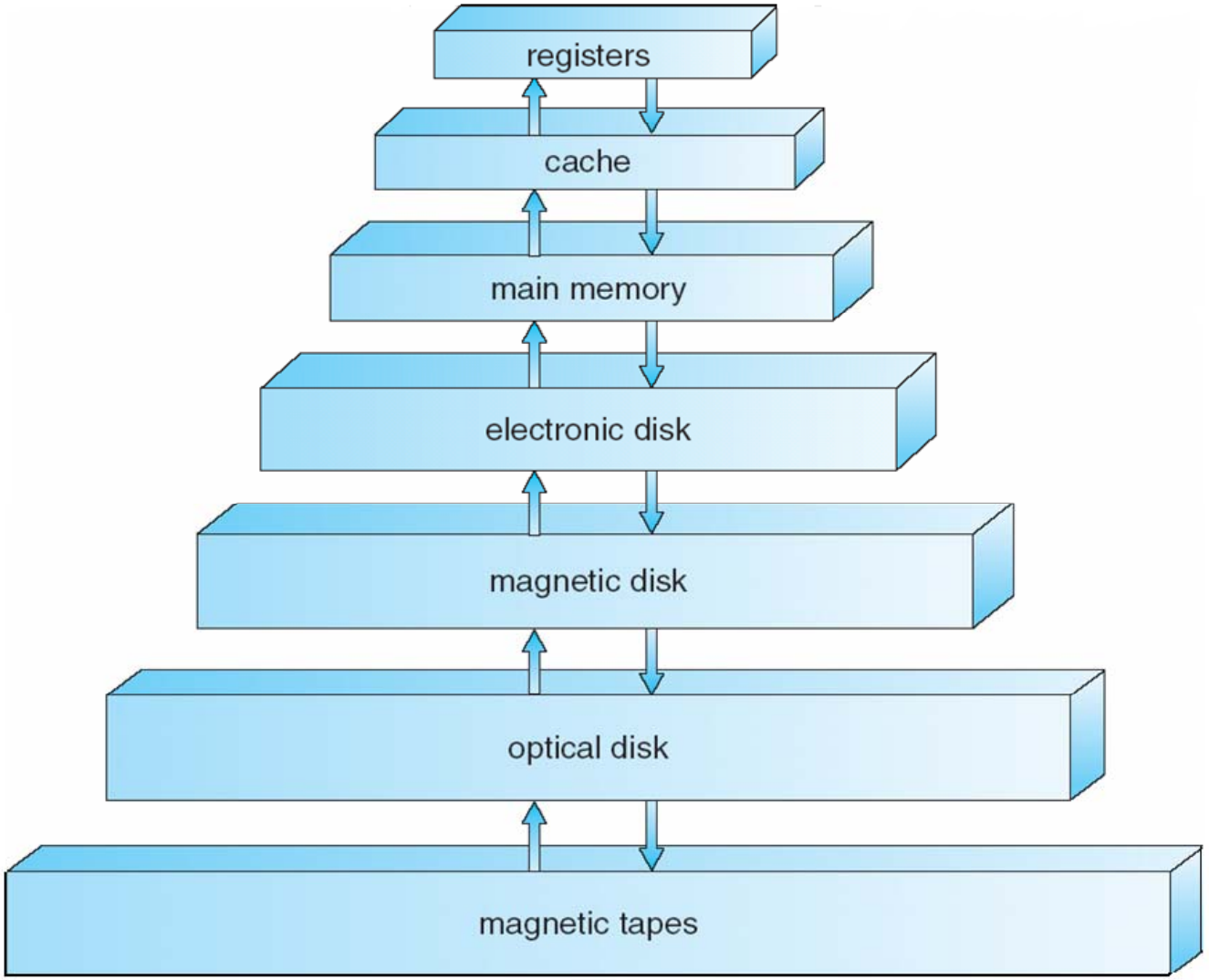
- Direct Memory Access (DMA)

- 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

Storage Structure

- Main memory – only large storage media that the CPU can access directly
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity
- Magnetic disks (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
- SSDs (Solid State Drive) use integrated circuit assemblies and interfaces, compatible with HDDs

Storage Device Hierarchy



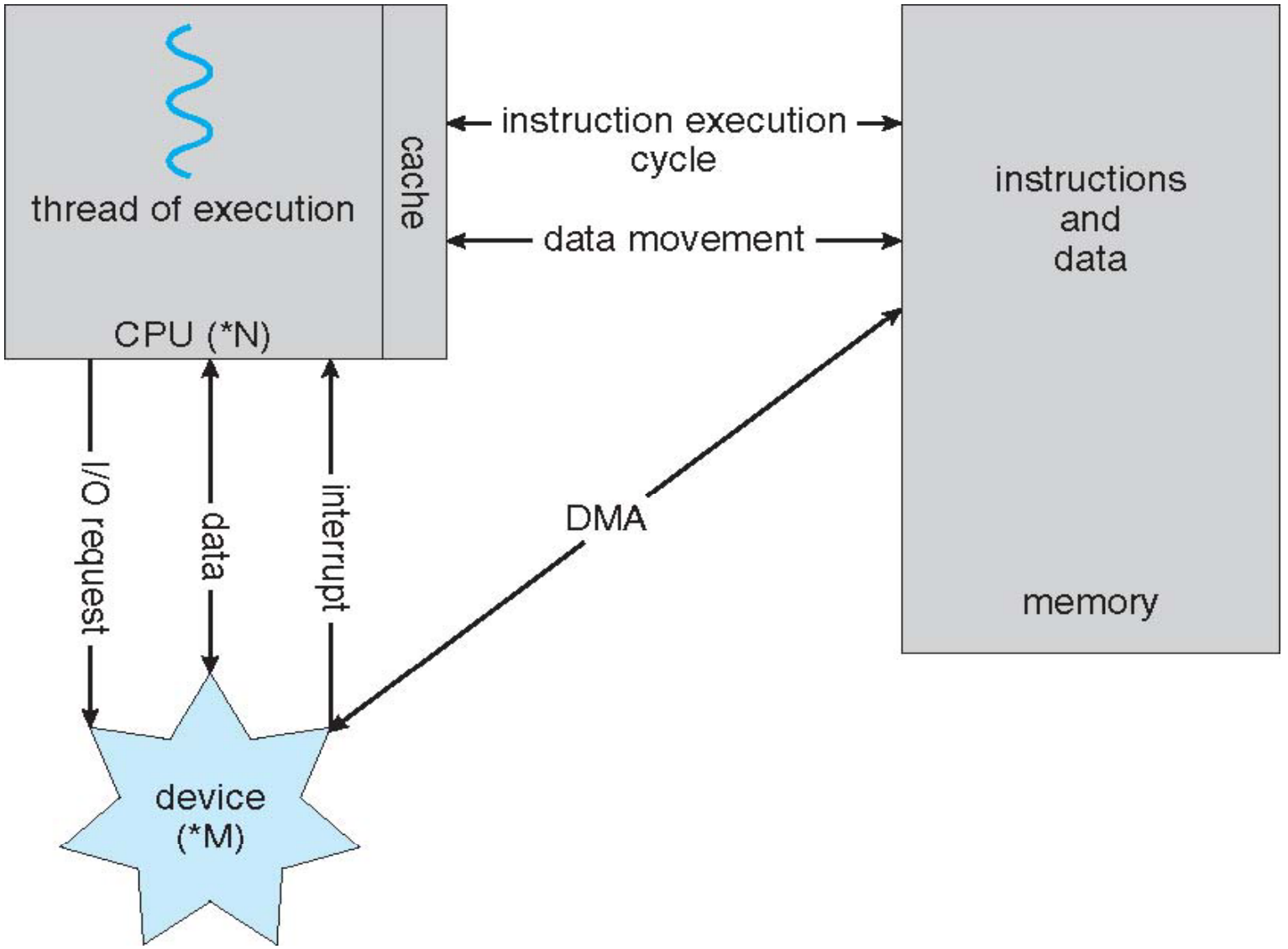
Caching

- 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

Computer System Architecture

- Most systems use one or more general-purpose processors
  - Most systems have special-purpose processors as well
- Multi-processor systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems
  - Advantages include
    - Increased throughput, economy of scale
    - Increased reliability – graceful degradation or fault tolerance
  - Two types
    - Asymmetric Multiprocessing
      - Only one or some CPUs may execute OS kernel code, I/O, use peripheral devices
    - Symmetric Multiprocessing

Modern Computer Operation



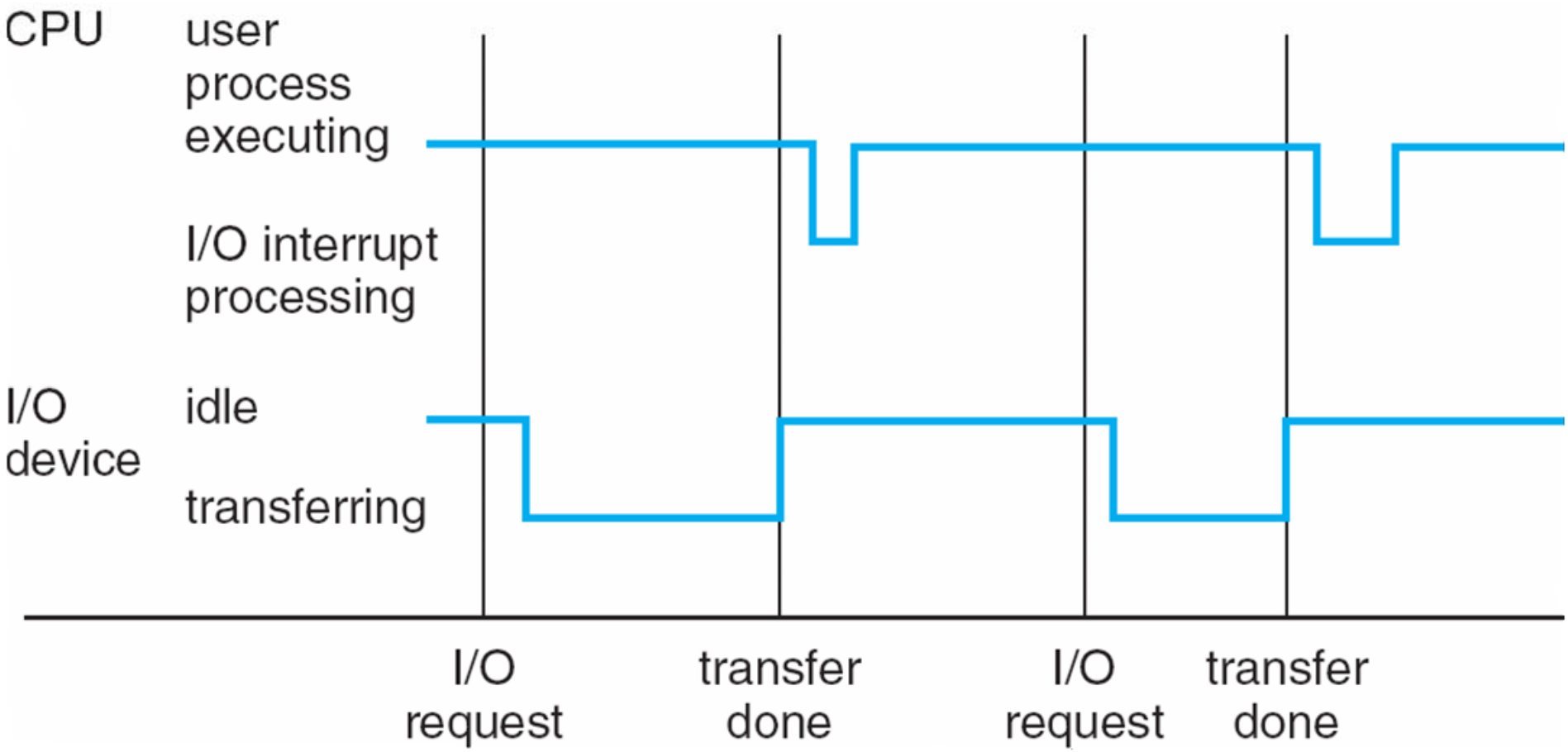
Interrupt

- An interrupt transfers control to the interrupt service routine generally, through the interrupt vector, which contains the addresses of all service routines
- The interrupt architecture must save the address of the interrupted instruction
- Incoming interrupts are disabled, while another interrupt is being processed to prevent a lost interrupt
- A trap is a software-generated interrupt caused either by an error or a user request
- Modern operating systems are interrupt-driven

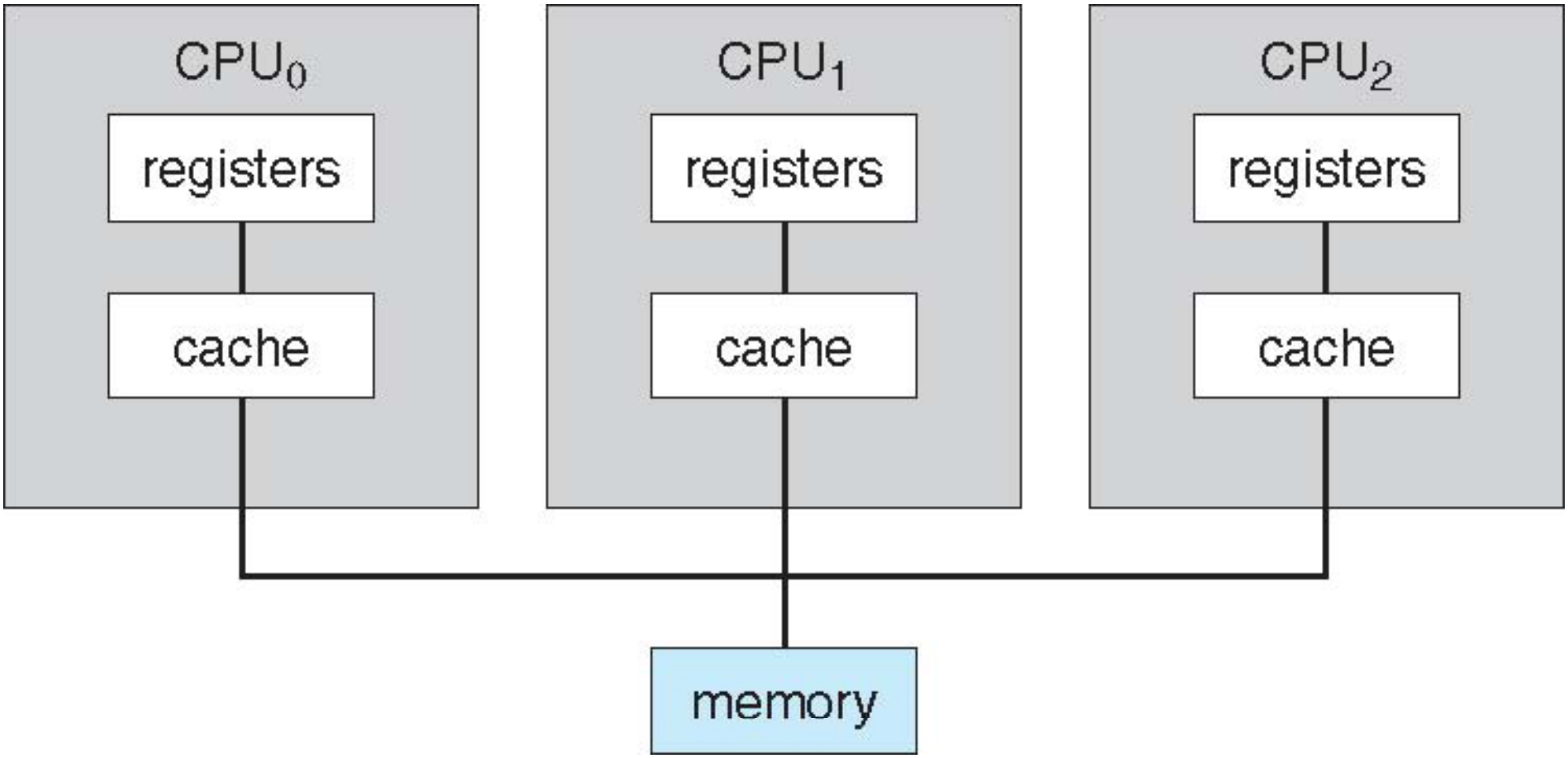
Interrupt Handling

- 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

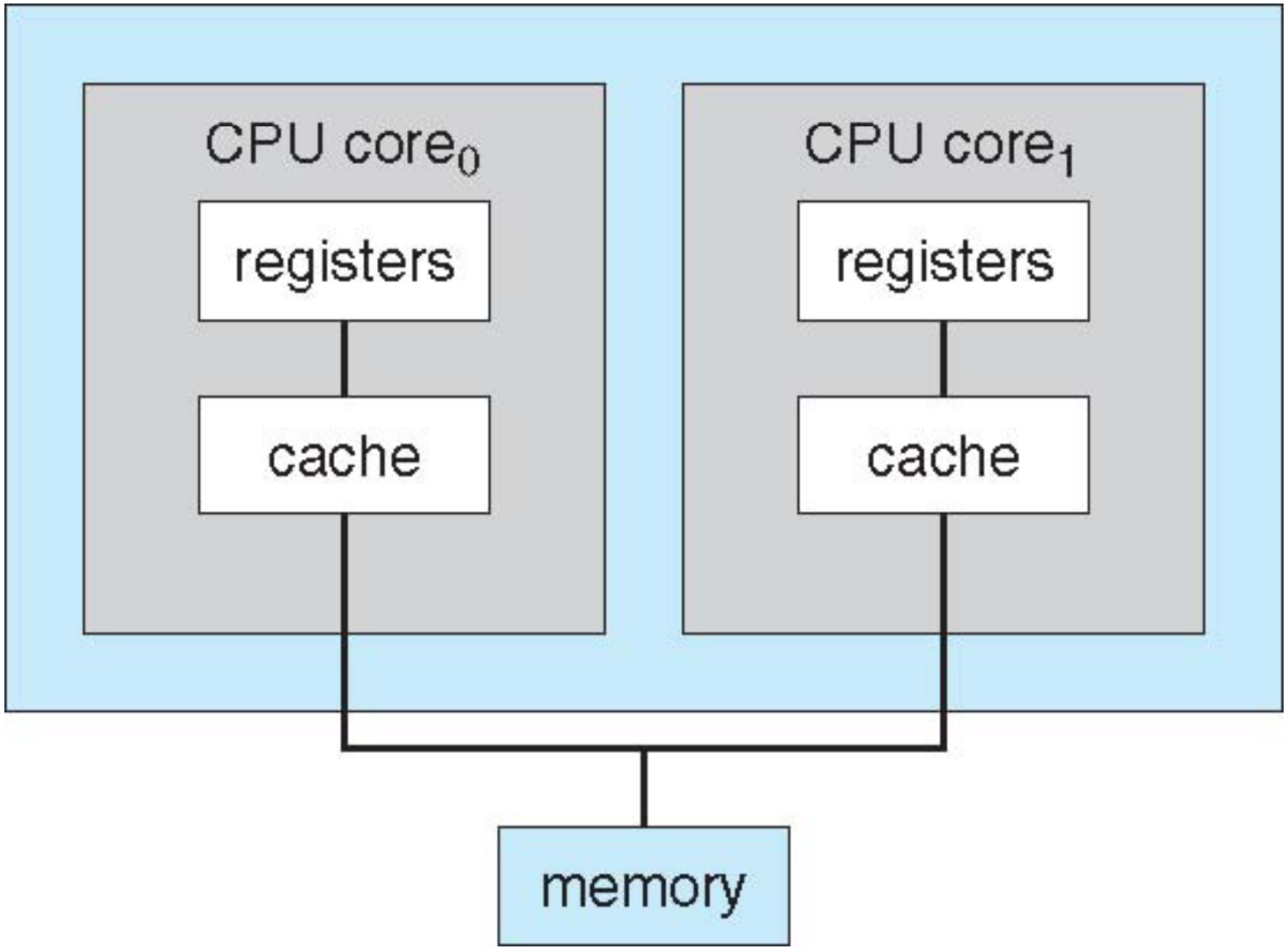
Interrupt Timeline



Symmetric Multi-processing Architecture (SMP)



A Dual Core Design



Clustered Systems

- Like multi-processor 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)
    - Applications must be written to use parallelization
  - Other clusters of computers are used for Web and cloud services

Operating System Structure (1)

- Multi-programming needed for efficiency
  - Single user cannot keep all CPUs, I/O devices both busy at all times
  - Multi-programming organizes jobs (code and data) so that 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
  - When job has to wait (*e.g.*, for I/O), OS switches to another job

Operating System Structure (2)

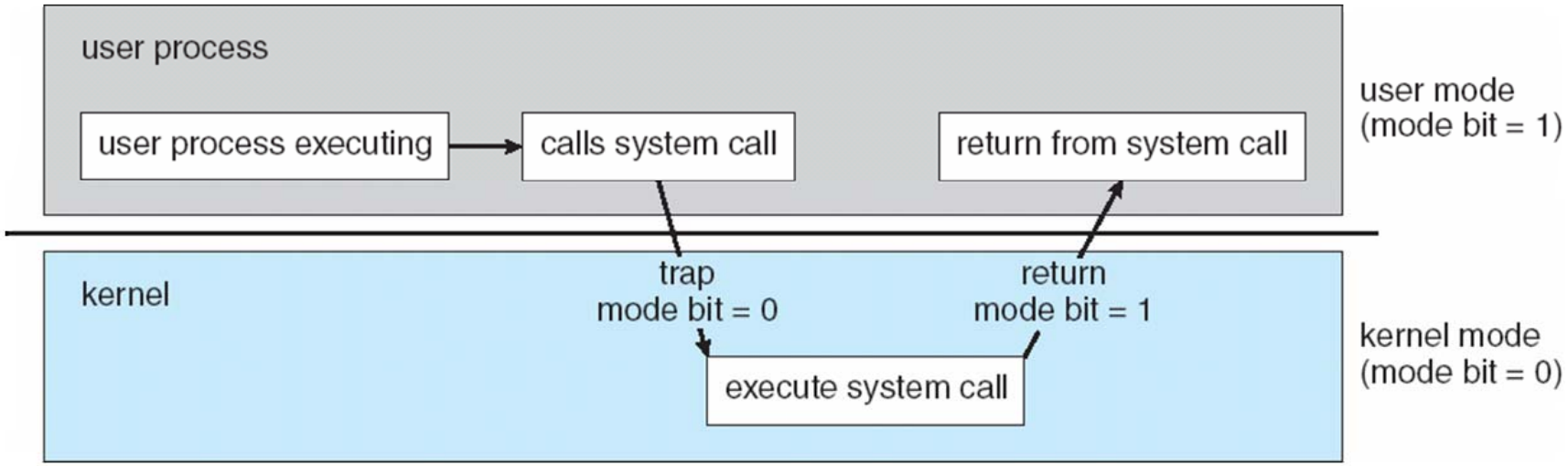
- Timesharing (multi-tasking) is the logical extension in which CPU switches jobs frequently that users can interact with each job while it is running
  → interactive computing
  - Response time should be below 1 second
  - User has at least one program executing in memory (process)
  - If several jobs ready to run at the same time
    → CPU scheduling
  - If processes do not fit in memory, swapping moves them in and out
  - Virtual memory allows execution of program not completely in memory

Operating System Operations

- Interrupt-driven by hardware
- Software error or request creates exception or trap
  - Division by zero, request for operating system service
  - Other process problems include infinite loops, processes modifying each other or the operating system
- 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, returns from call resets to user

Transition from User to Kernel Mode

- Timer to prevent infinite loop/process hogging resources
  - Set interrupt after specific period
  - Operating system decrements counter
  - When counter equals zero interrupt is generated
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Process Management (1)

- 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, executing the program instructions
  - CPU, memory, I/O, files
  - Initialization data

- Process termination requires reclaim of any reusable resources

Process Management (2)

- 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 a system has many processes, some users, some operating systems running concurrently on one or more CPUs
  - Concurrency is reached by multiplexing CPUs among processes and threads

Process Management Activities

- The operating system is responsible for the following activities in connection with the 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

Memory Management

- Prerequisites for processes to run:
  - All data in memory before and after processing
  - All instructions in memory in order to execute
- 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 de-allocating memory space as needed

Storage Management (1)

- 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)

Storage Management (2)

- 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 (1)

- 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

Mass Storage Management (2)

- 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
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

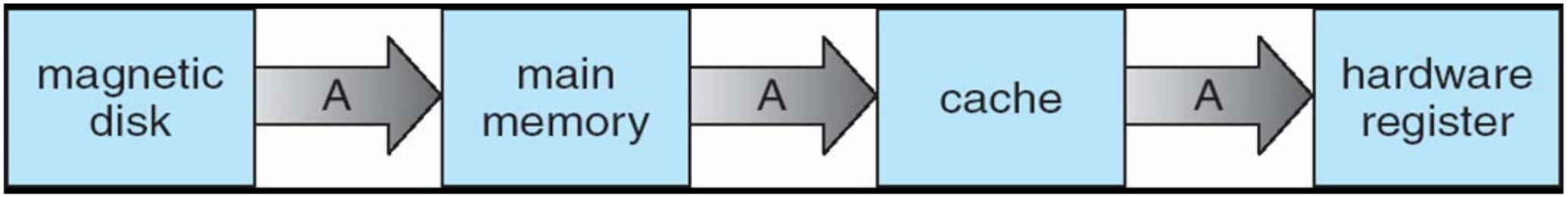
Performance of Various Levels of Storage

- Movement between levels of storage hierarchy can be explicit or implicit

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk SSD
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

Migration of Integer *A* from Disk to Register

- Multi-tasking environments must be careful to use most recent values, no matter where it is stored in the storage hierarchy



- Multi-processor environments must provide cache coherency in hardware such that all CPUs have the most recent values in their cache
- A distributed environment situation even more complex
  - Several copies of a datum can exist
  - Various solutions covered in Distributed File Systems

I/O Subsystem

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

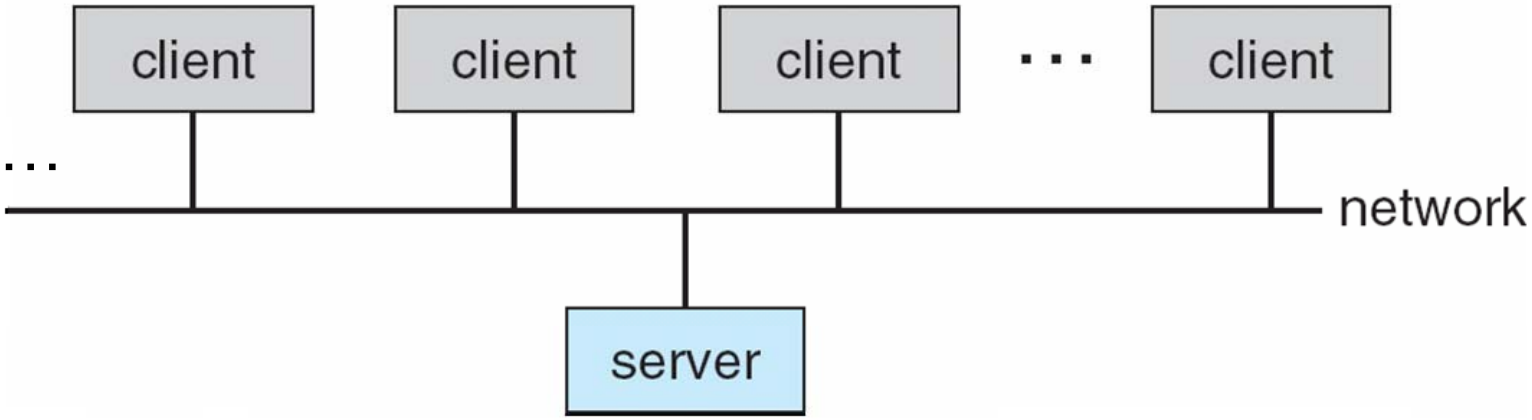
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices

Protection and Security

- 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 and a 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 (IDs) include one name, associated number
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (ID) allows set of users to be defined and controls managed, also associated with each process, file
  - Privilege escalation allows user to change to effective ID (more rights)

Distributed Computing

- Client/Server Computing
- Many systems are now servers, responding to requests generated by clients
  - Compute server provides an interface to client to request services (i.e. database)
  - File server provides interface for clients to store and retrieve files
  - Web servers …



Peer-to-Peer Computing

- Peer-to-Peer (P2P) another model of distributed system

- P2P does not distinguish between 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*

Web-based Computing

- The Web has become ubiquitous
- PCs are most prevalent devices in the past
  - Now many smart mobile devices exist
- More (basically all) devices becoming networked to allow Web (or server) access
- Use of operating systems like Windows 95, client-side only, have evolved into Linux, Windows 10, and Mac OS X, which can be used both for clients and servers

Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary closed-source
- Counter move to the copy protection and Digital Rights Management (DRM) movement
- Started by Free Software Foundation (FSF), which has "copyleft" GNU Public License (GPL)
  - Examples include GNU/Linux, BSD UNIX (including core of Mac OS X), and Sun Solaris