CECS 326
Operating Systems

1. Overview

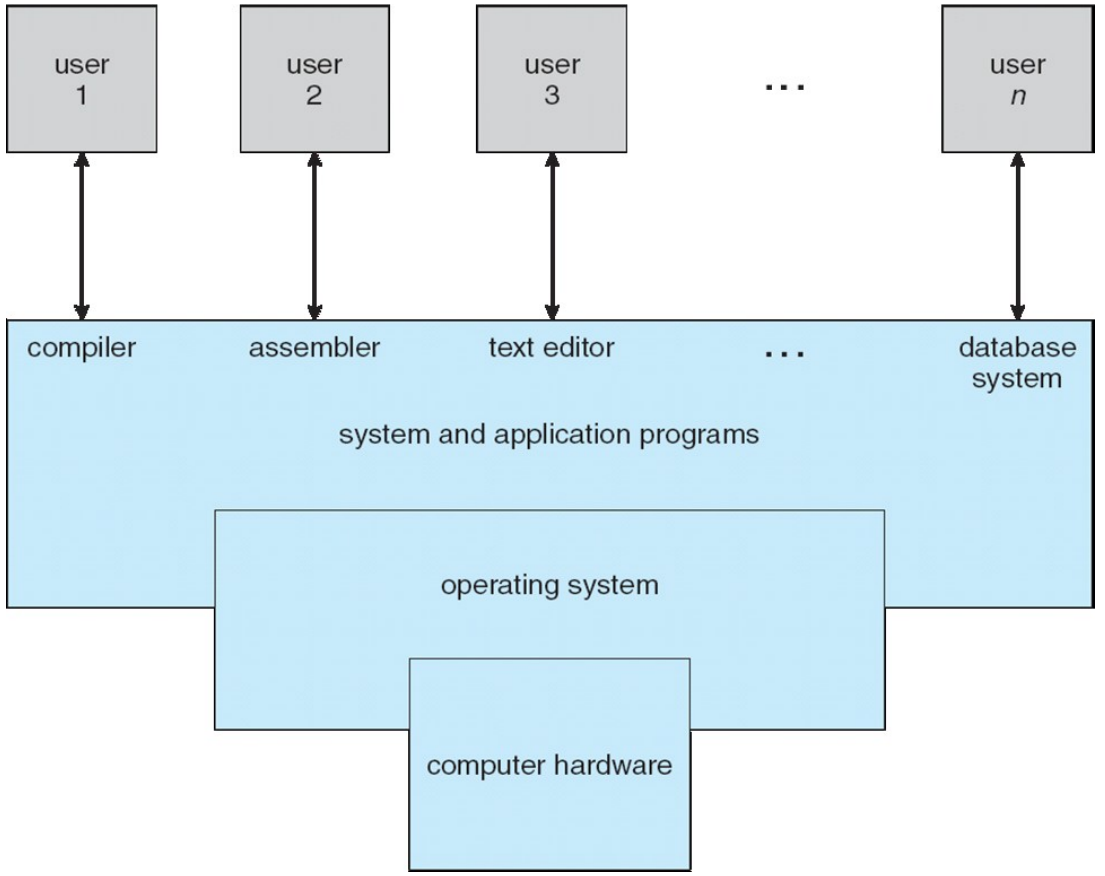
(Reference: Operating system Concepts by Silberschatz, Galvin and Gagne)

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

- Consists of 4 components:
  - Hardware – provides basic computer resources (e.g., CPU, memory, I/O devices)
  - Operating system – controls and coordinates use of hardware among various applications
  - Applications – define the ways in which the system resources are used to solve the computer problems of the users (e.g., word processors, compilers, web browsers, database systems, video games)
  - Users – people, machines, other computers



What Do Operating Systems Provide?

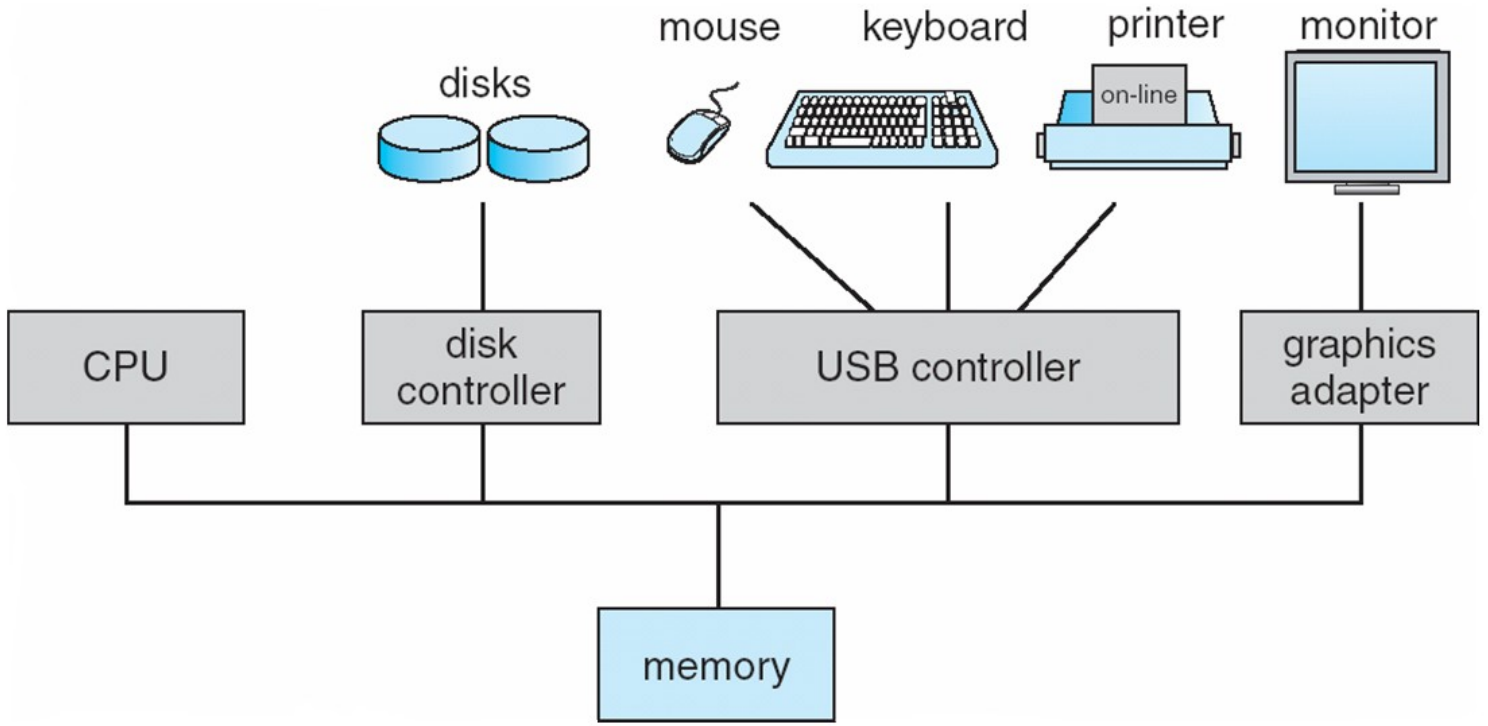
- Depends on the point of view
- Users want convenience, ease of use, and good performance. Don’t care about resource utilization
- Shared computers such as mainframe and minicomputers must keep all users happy
- Dedicated systems such as workstations have dedicated resources but frequently use shared resources from servers
- Handheld computers are resource poor, need to be optimized for usability and battery life
- Some computers have little or no user interface, e.g., embedded computers in devices and automobiles

Operating System Definition

- No universally accepted definition
- Some view it as a resource manager
  - Allocates/deallocates all resources
  - Decides between conflicting requests for efficient and fair use of resources
- Others view it as a control program
  - Controls execution of programs to prevent errors and improper use of the computer
- The part of OS running at all times on the computer is the kernel
- The remaining parts of the OS are referred to as a system programs

Computer System Organization

- 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



What happens at Computer Startup

- Bootstrap program is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as firmware
  - Initializes all aspects of system, including CPU registers, device controllers, and memory contents
  - Loads operating system kernel and starts execution
  - Some system programs can also be loaded at boot time to become system processes or system daemons
- Once system is fully booted, it can start providing services in response to events signaled by interrupts to the CPU from hardware or software

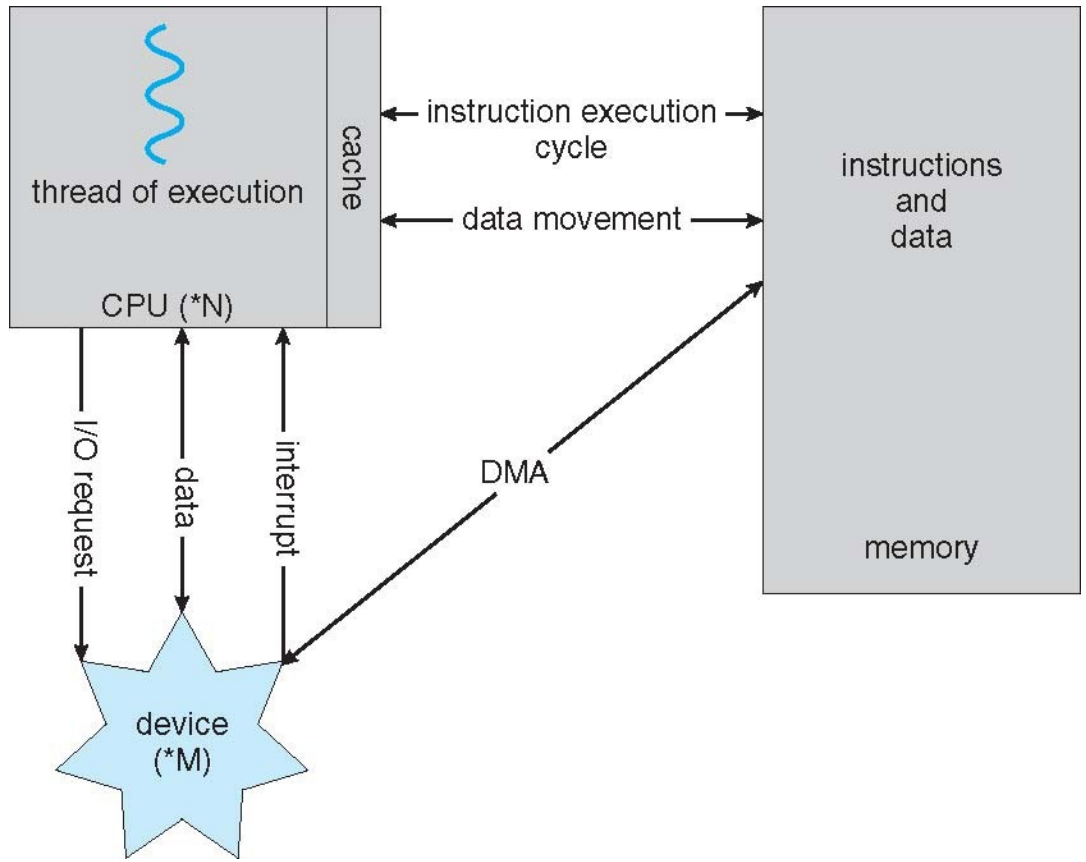
Computer System Operation

- 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, and CPU moves data from/to main memory to/from these 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

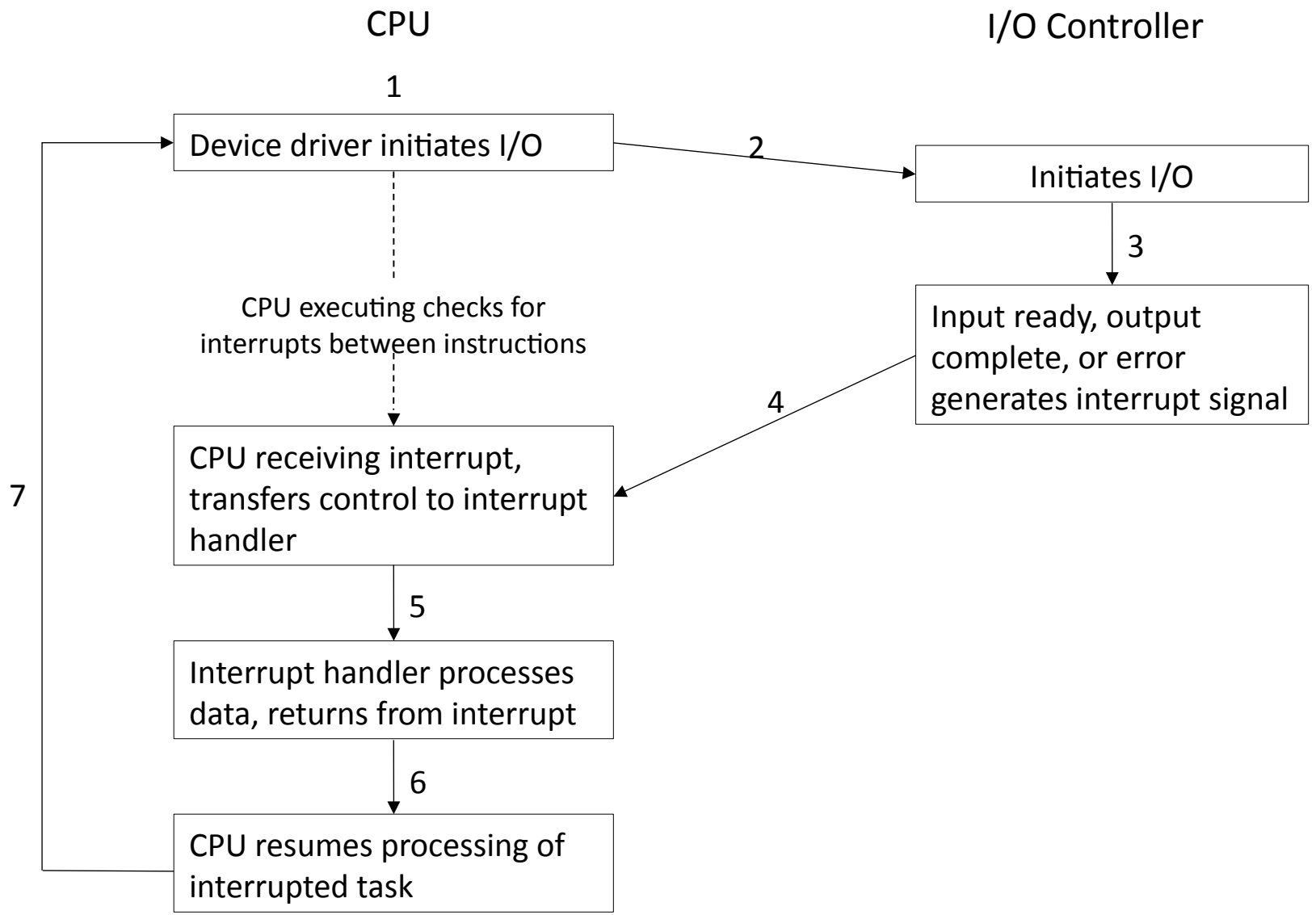
Interrupts and Interrupt Handling

- When CPU is interrupted, it stops what it is doing and immediately transfers execution to an interrupt service routine (interrupt handler) generally, through the interrupt vector, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction so that interrupted computation can resume upon completion of the interrupt service routine
- A trap or exception is a software-generated interrupt caused either by an error or a user request
- Operating systems are interrupt driven
  - OS preserves state of CPU by storing registers and program counter
  - Determines which type of interrupt has occurred through polling of a vectored interrupt system
  - Different code performed action defined for different type of interrupt

Interactions Among Components



Interrupt-driven I/O Cycle



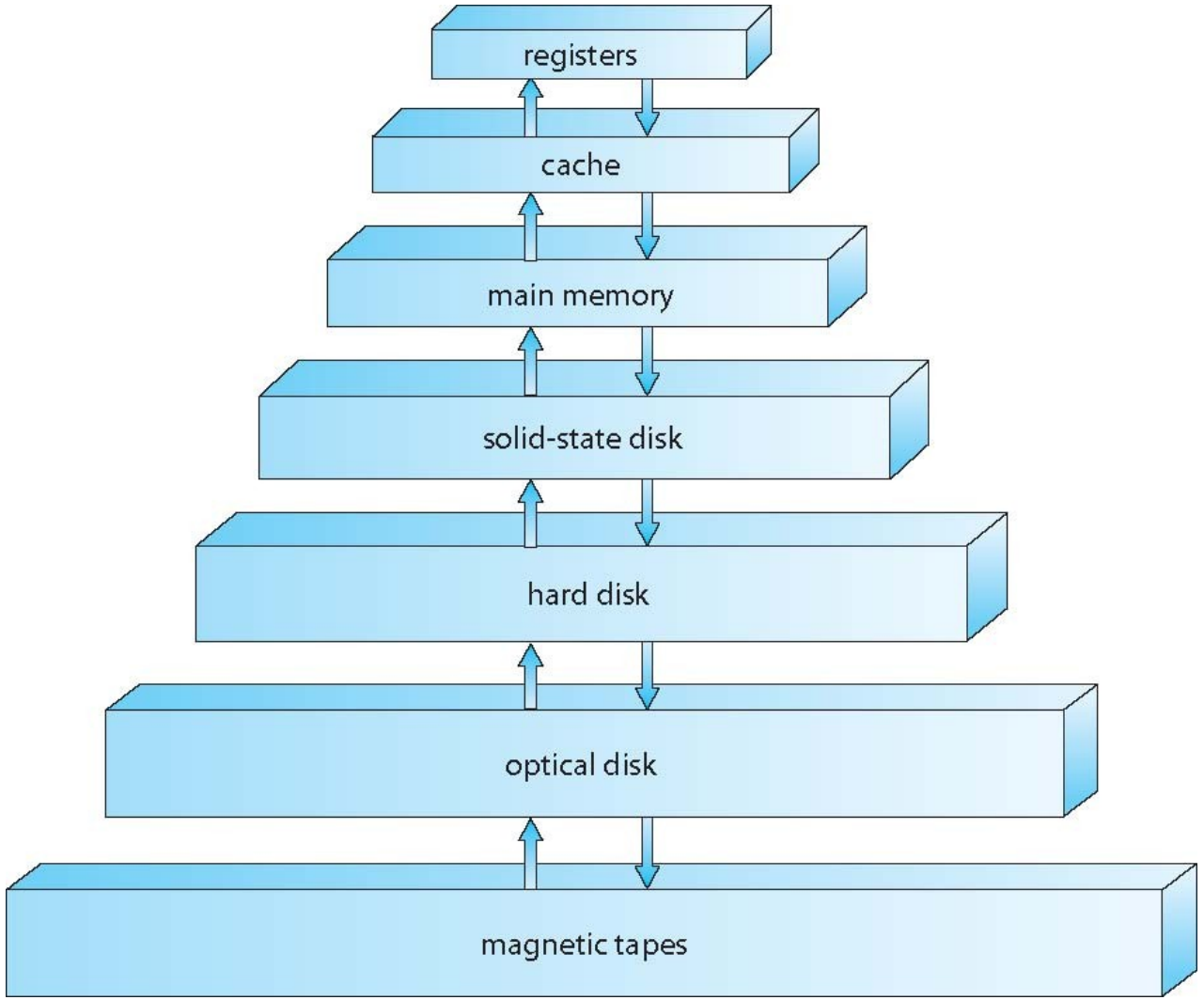
Storage Structure

- Main memory – the only large storage media that CPU can access directly
  - Random access
  - Typically volatile
- 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 logically divided into tracks, which are subdivided into sectors
    - Disk controller determines the logical interaction between the device and the computer
  - Solid-state disks – faster than hard disks, nonvolatile
    - Various technologies
    - Becoming more popular

Storage Hierarchy

- Storage systems are organized in hierarchy
  - Speed
  - Cost
  - Volatility
- Caching – copying information into faster storage system
  - An important principle used at many levels (hardware, software, operating system)
  - Information in use is copied from slower to faster storage temporarily
  - Faster storage (cache) is checked first to determine if information is there; use from there if yes, otherwise data is copied to cache and used there
  - Cache is smaller than storage being cached
    - Cache management is an important design problem
    - Need to consider cache size and replacement policy
- Device driver for each device controller to manage I/O to provide uniform interface between controller and kernel

Storage Device Hierarchy



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

Movement between levels of storage hierarchy can be explicit or implicit

Storage Definitions and Notation Review

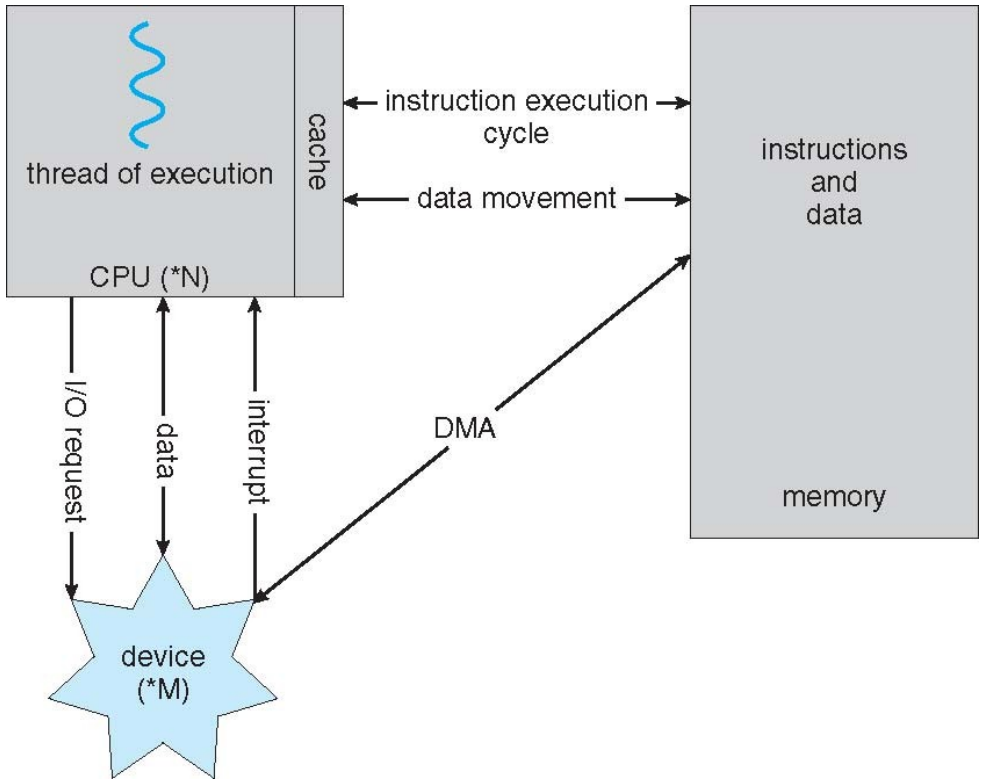
- The basic unit of computer storage is the **bit**. A bit can contain one of two values, 0 and 1. All other storage in a computer is based on collections of bits. Given enough bits, it is amazing how many things a computer can represent: numbers, letters, images, movies, sounds, documents, and programs, to name a few. A **byte** is 8 bits, and on most computers it is the smallest convenient chunk of storage. For example, most computers don't have an instruction to move a bit but do have one to move a byte. A less common term is **word**, which is a given computer architecture's native unit of data. A word is made up of one or more bytes. For example, a computer that has 64-bit registers and 64-bit memory addressing typically has 64-bit (8-byte) words. A computer executes many operations in its native word size rather than a byte at a time.

- Computer storage, along with most computer throughput, is generally measured and manipulated in bytes and collections of bytes.
- A **kilobyte**, or **KB**, is 1,024 bytes
- a **megabyte**, or **MB**, is $1{,}024^2$ bytes
- a **gigabyte**, or **GB**, is $1{,}024^3$ bytes
- a **terabyte**, or **TB**, is $1{,}024^4$ bytes
- a **petabyte**, or **PB**, is $1{,}024^5$ bytes

- Computer manufacturers often round off these numbers and say that a megabyte is 1 million bytes and a gigabyte is 1 billion bytes. Networking measurements are an exception to this general rule; they are given in bits (because networks move data a bit at a time).

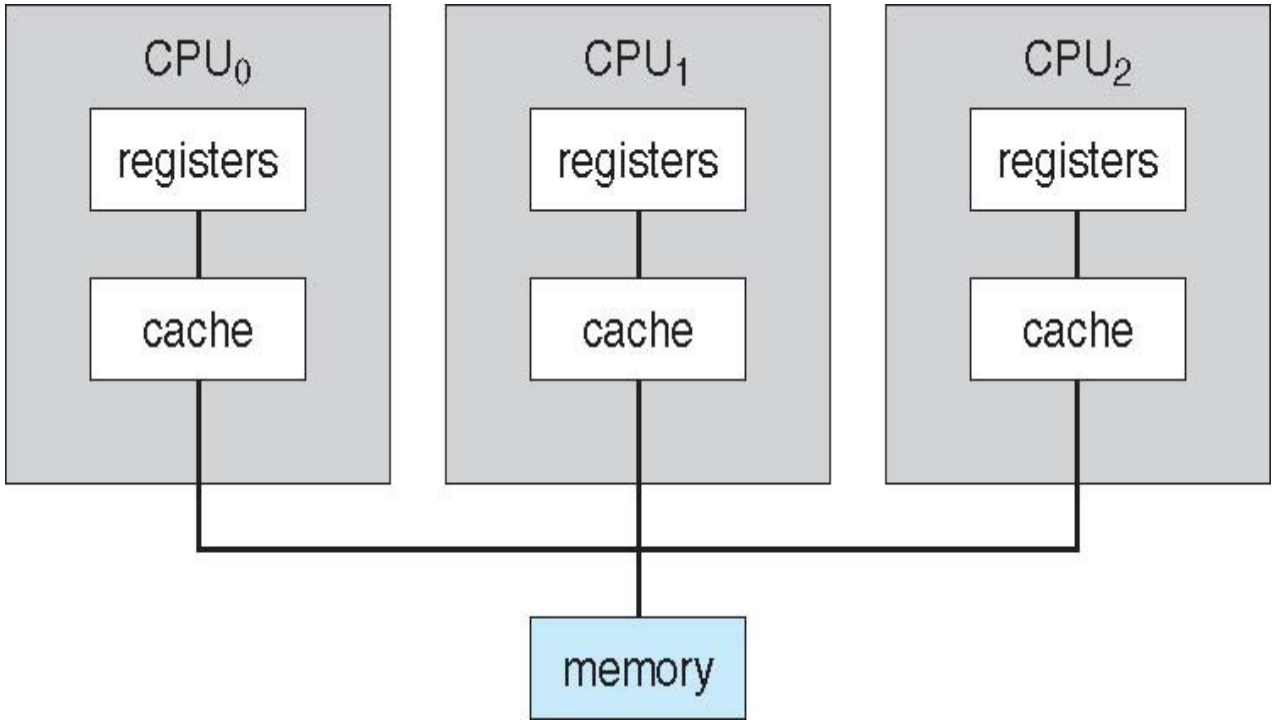
Direct Memory Access (DMA) Structure

- Used for high-speed I/O devices able to transmit data 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 one interrupt per byte



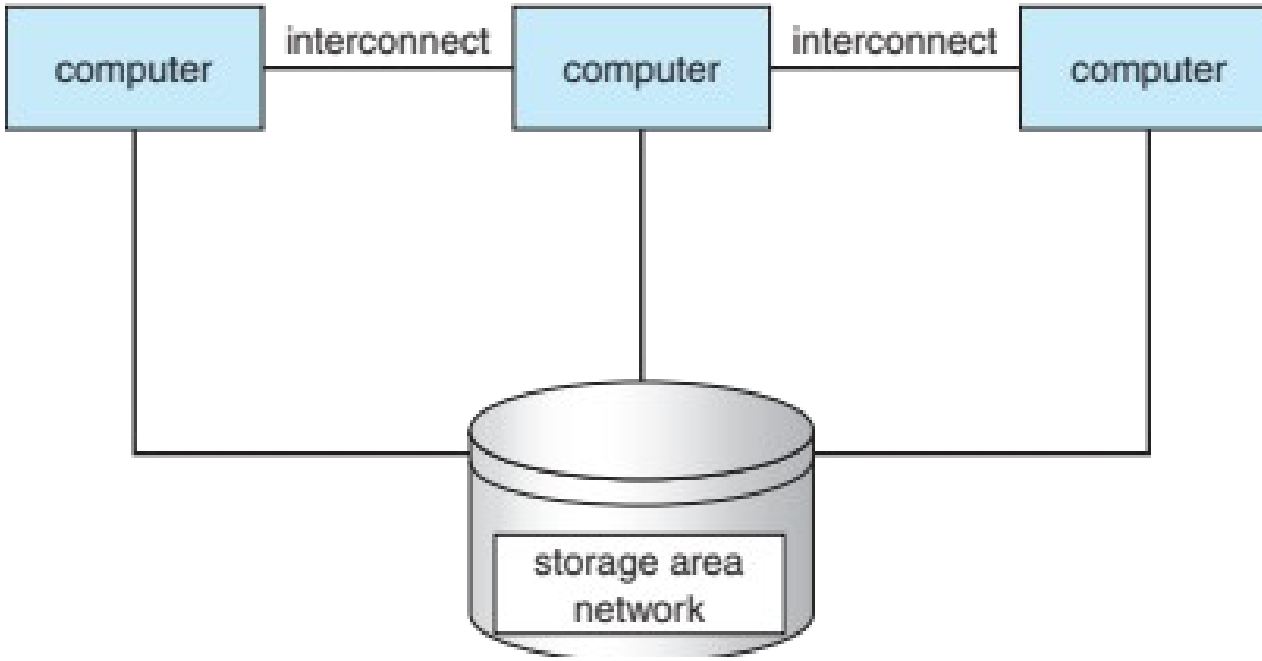
Computer-System Architecture

- Single-Processor System
- Multiprocessor Systems
  - Growing in use and importance
  - Advantages include:
    - Increased throughput
    - Economy of scale
    - Increased reliability, providing graceful degradation or fault tolerance



Computer-System Architecture

- Clustered Systems
  - Like multiprocessor systems, but multiple computers working together
  - This setup is often referred to as loosely coupled, as opposed to tightly coupled multiprocessing
    - Usually sharing storage via a storage area network (SAN)
    - Provides 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

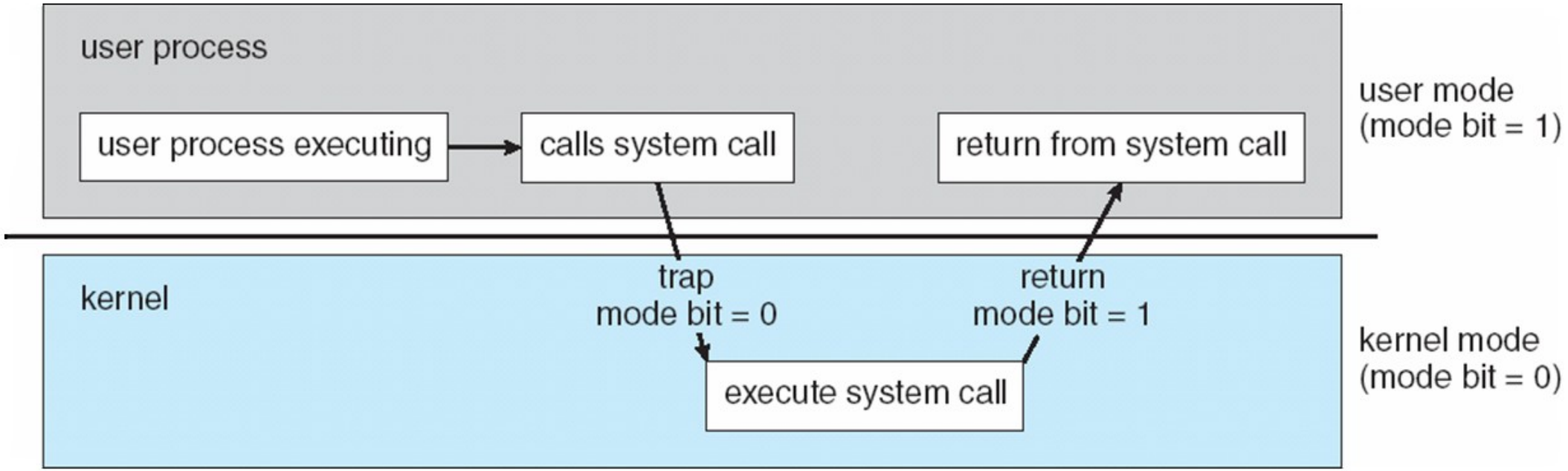


Operating System Operations

- Interrupt driven (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (exception or trap)
    - Software error (e.g., division by zero) and other process problems (e.g., processes modifying each other or the OS)
    - Request for OS 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 mode
- Some CPUs support multi-mode operations (e.g., virtual machine mode, or VMM, used for virtualization management software)

Transition from User to Kernel Mode

- Timer is used to prevent infinite loop or process hogging resources
  - Timer is set to interrupt the computer after some time period, which can be fixed or variable. Instruction that modifies timer is privileged.
  - Variable timer usually implemented with a fixed-rate clock and a counter
  - Counter is decremented at each clock tick, interrupt when zero
  - Timer is set up before turning control to a user process
  - When timer interrupts, control transfers automatically to the OS
- Mode transition can also occur due to a trap via system call



Functions Provided by OS for Workload & Resource Management

- Process Management
- Memory Management
- File System Management
- Mass Storage Management
- I/O System Management

Process Management

- A process is a program in execution. It is a unit of work that the OS manages. 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 all reusable resources
- Single-threaded process has one program counter specifying address of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter for each thread
- Typically system has many processes, some user and some OS, running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes/threads

Process Management Activities

- OS 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

Memory Management

- To execute a program, all (or part) of the instructions and needed data must be in memory
- Memory management determines what is in memory, where and when
  - Purpose is to optimize 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 part thereof) 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
  - Abstract physical properties to logical storage unit – file
  - Each medium is controlled by device (e.g., 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 and directories
    - Primitives to manipulate files and directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

Mass Storage Management

- Mass-storage management
  - Disks usually used to store data that does not fit in main memory (virtual memory) or for long-term storage
  - Proper management is of central importance
  - Entire speed of computer operation hinges on disk subsystem and its efficiency
  - 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
    - Varies between WORM (write-once, read-many-times) and RW (read-write)

I/O Subsystem

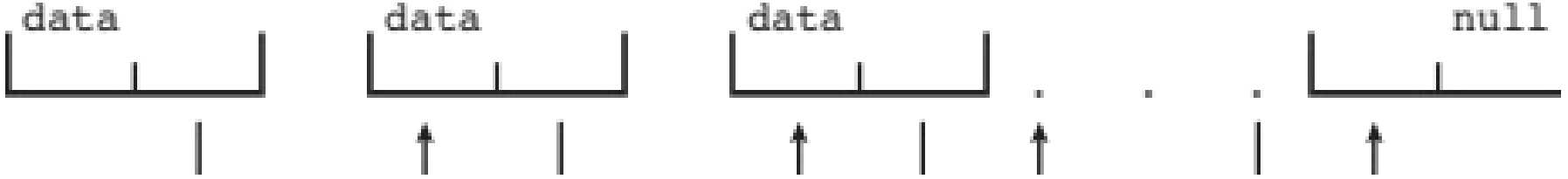
- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem is responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (e.g., print spooling)
  - General device-driver interface
  - Drivers for specific hardware devices

Protection and Security

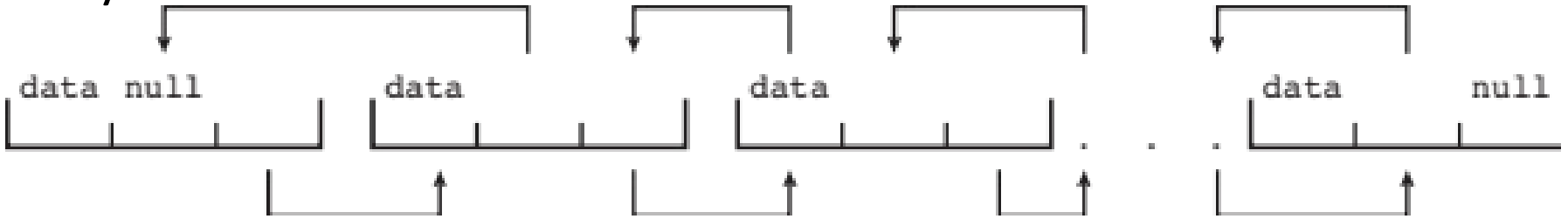
- Protection – any mechanism for controlling access of processes or users to resources defined by the OS
- Security – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users to determine who can do what
  - User identities (user IDs, 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 identified and controls managed, then also associated with each process and file

Kernel Data Structures

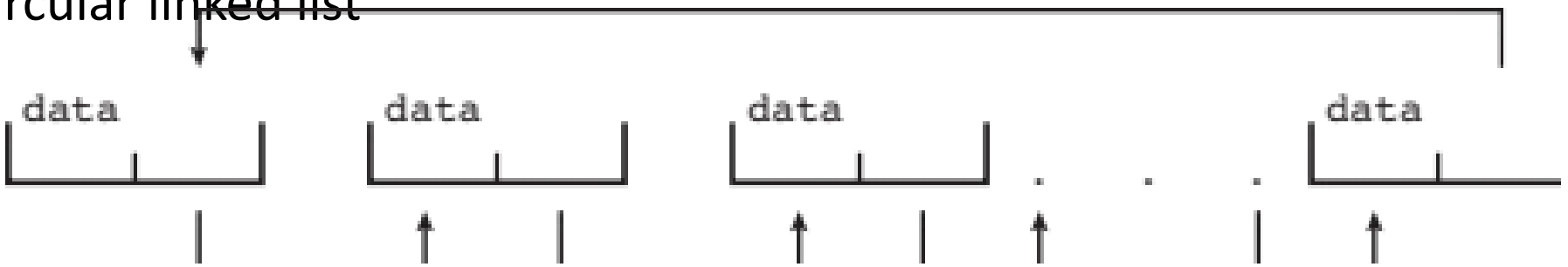
- Many similar to standard programming data structures (linked lists, stack, queue)
  - Singly linked list



  - Doubly linked list

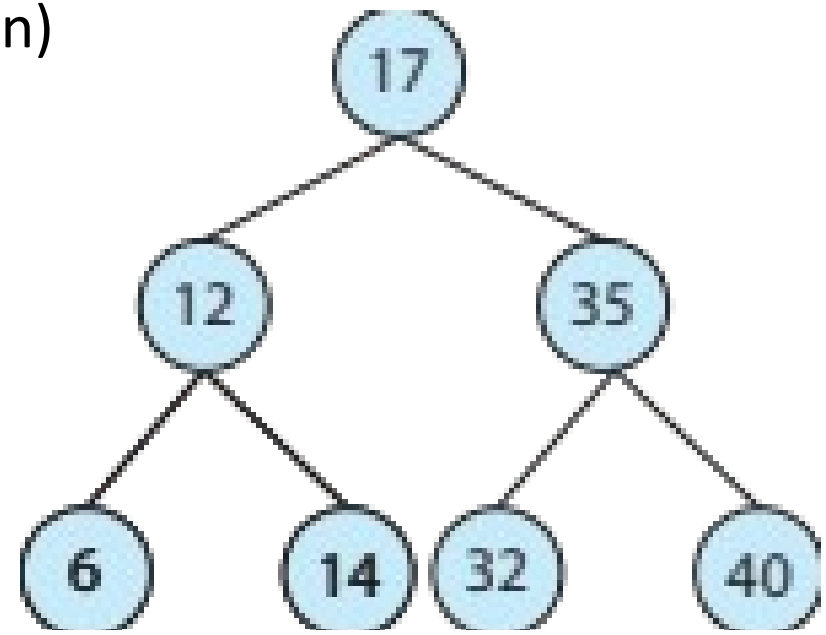


  - Circular linked list



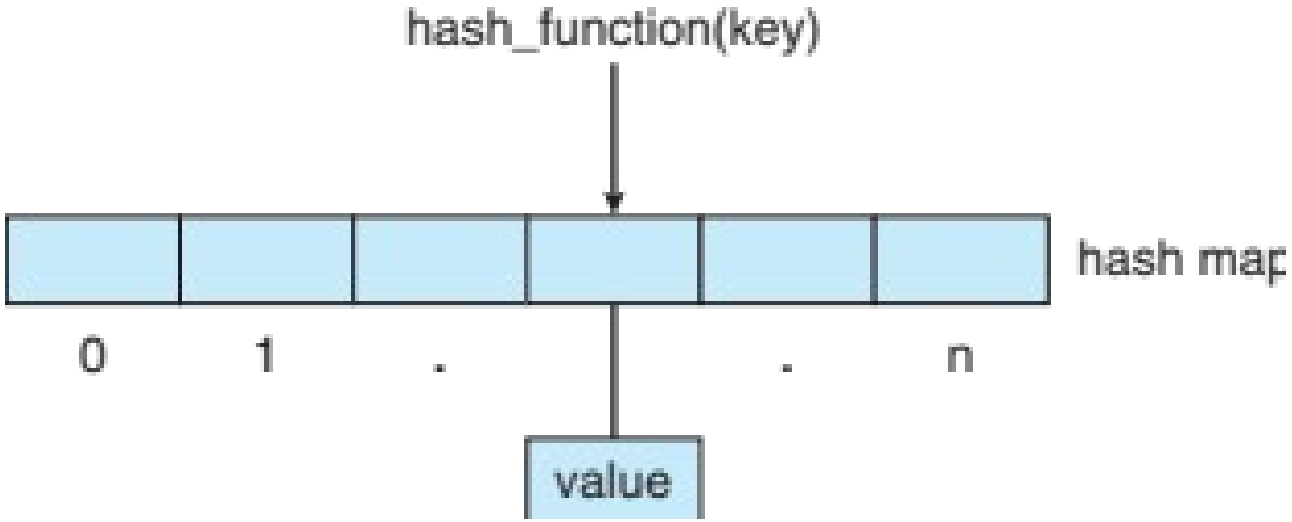
Kernel Data Structures

- Binary search tree (left <= right)
  - Worst-case search performance is O(n)
- Balanced binary search tree
  - Search performance is O(log 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 are defined in include files <linux/list.h>, <linux/kfifo.h>, <linux/rbtree.h>