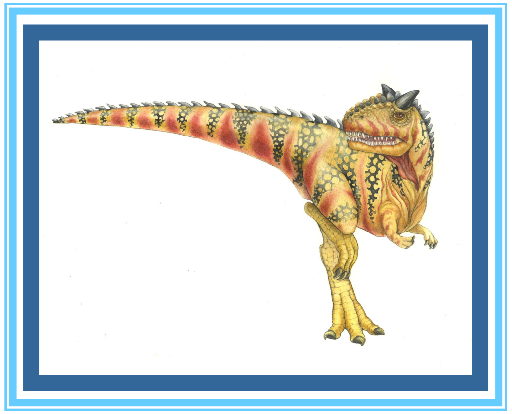
Chapter 1: Introduction



Chapter 1: Introduction

- 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
- Kernel Data Structures
- Computing Environments
- Open-Source Operating Systems

Objectives

To describe the basic organization of computer systems

To provide a grand tour of the major components of operating systems

To give an overview of the many types of computing environments

To explore several open-source operating systems

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:

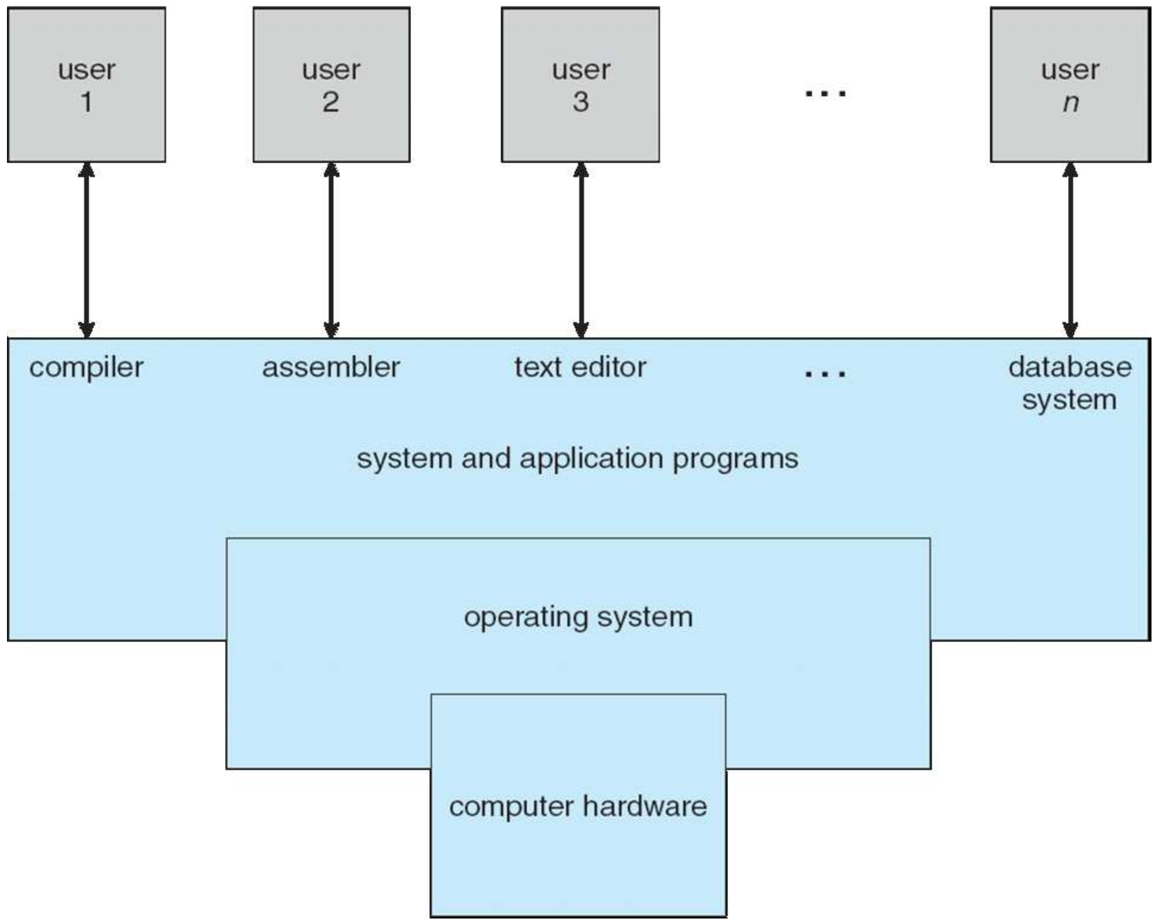
- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

Computer System Structure

Computer system can be divided into four components:

- Hardware – provides basic computing resources
  - CPU, memory, I/O devices
- Operating system
  - Controls and coordinates use of hardware among various applications and users
- 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
- Users
  - People, machines, other computers

Four Components of a Computer System



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 dedicated 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

Operating System Definition

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 (Cont.)

No universally accepted definition

"Everything a vendor ships when you order an operating system" is good approximation

- But varies wildly

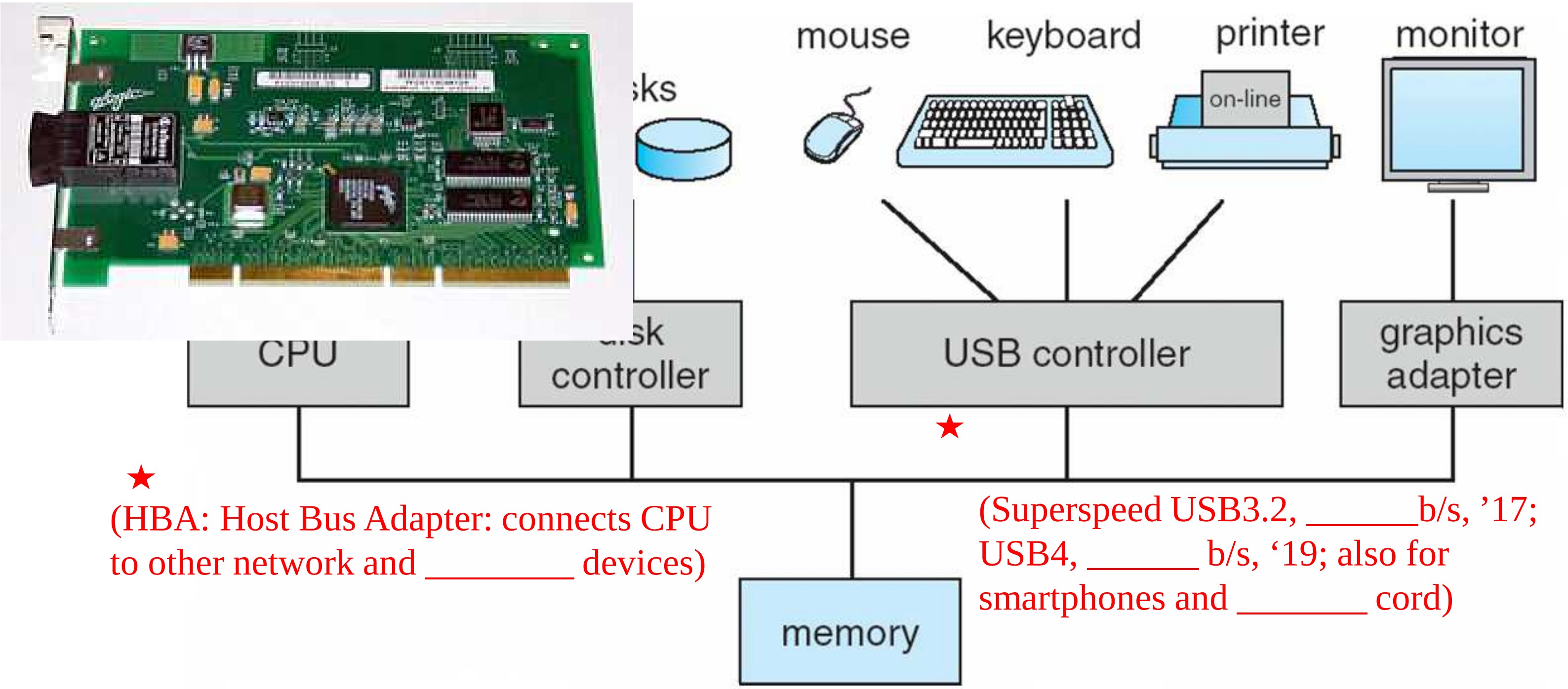
"The one program running at all times on the computer" is the **kernel**. Everything else is either a system program (ships with the operating system) or an application program.

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
  - Loads operating system kernel and starts execution

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



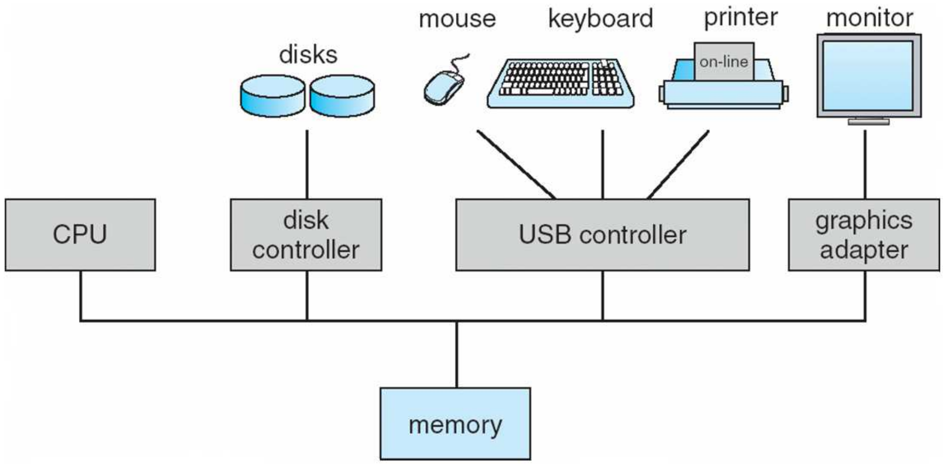
★ (HBA: Host Bus Adapter: connects CPU to other network and ________ devices)

★ (Superspeed USB3.2, ______b/s, ’17; USB4, ______ b/s, ‘19; also for smartphones and _______ cord)

Computer-System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
  ★(HW converting serial bit streams into ______ ______ and ______ correction)
- Each device controller has a local 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★(by HW (anytime) or SW (system call))

★



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 ★(in stack)
- 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

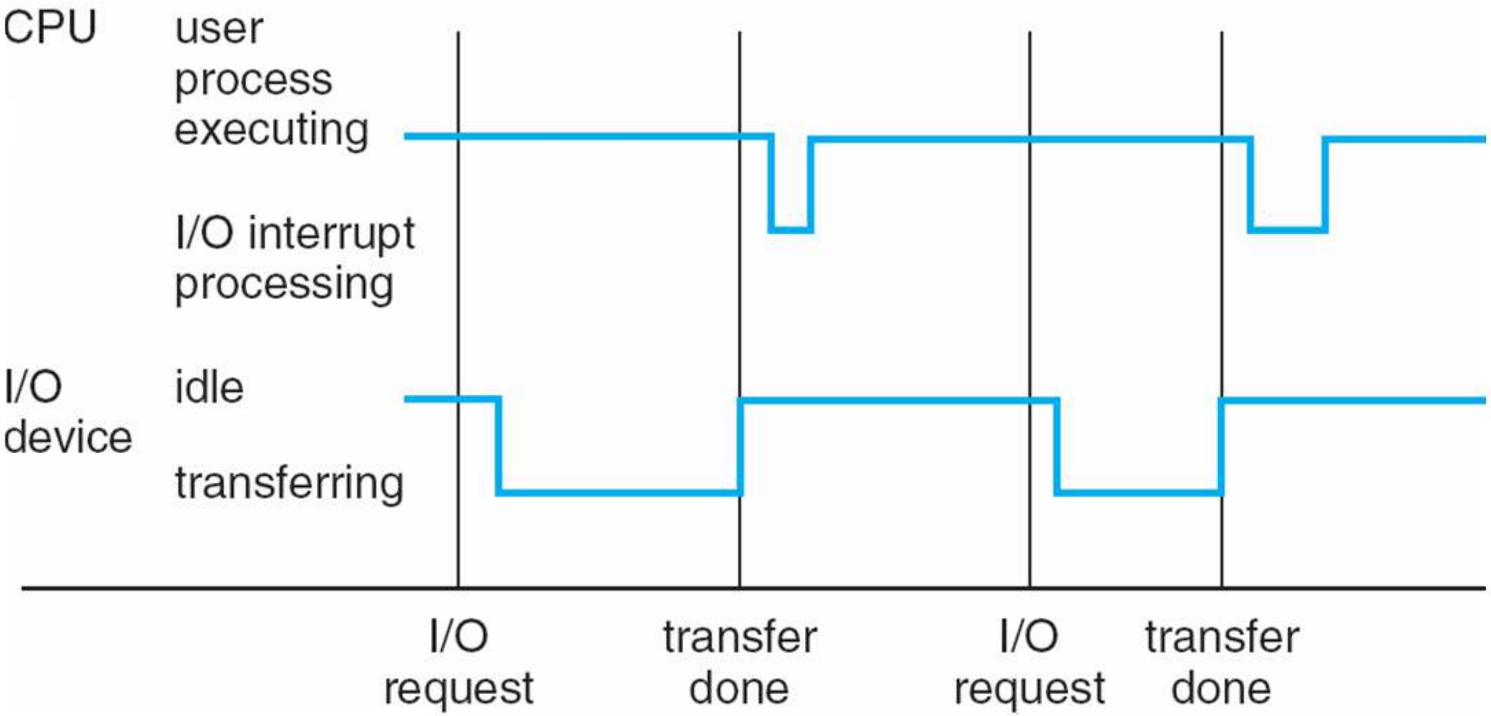
The operating system preserves the state of the CPU by storing registers and the program counter

Determines which type of interrupt has occurred:

- **polling**★(slow)
- **vectored** interrupt system★(fast and most OSs)

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

Interrupt Timeline



I/O Structure

★(Synchronous I/O)
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) ★(Loop: jmp Loop)
- At most one I/O request is outstanding at a time, no simultaneous I/O processing

★(Asynchronous I/O)
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 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 1,0242 ytes; a **gigabyte**, or **GB**, is 1,0243 bytes; a **terabyte**, or **TB**, is 1,0244 bytes; and a **petabyte**, or **PB**, is 1,0245 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).

★(____ 10^{18} (2^{60}); ______ 10^{21} (2^{70}); ______ 10^{24} (2^{80});

★(deci (10^{-1}); centi; milli; micro; _____; _____; ______; _____; ______; ______ (10^{-24}))

Direct Memory Access Structure

Used for high-speed I/O devices able to transmit information at close to memory speeds★ (DMA Chip no. : ______)

★ (block size: 128B ~ 4KB)

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 (burst transfer for _____ peripheral; _____ ______ for slow one)

★(memory-mapped I/O vs _____-mapped I/O)

★(___________)) I/O (continuous polling of status bit of device controller) vs ________ _____ I/O)

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 **nonvolatile** storage capacity
- Magnetic 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** – faster than magnetic disks, nonvolatile
  - Various technologies★ (ReRAM or RRAM (_______ RAM): non-volatile RAM; __B scale; not yet replacement of flash memory)
  - Becoming more popular

Storage Hierarchy

Storage systems organized in hierarchy

- Speed
- Cost
- Volatility

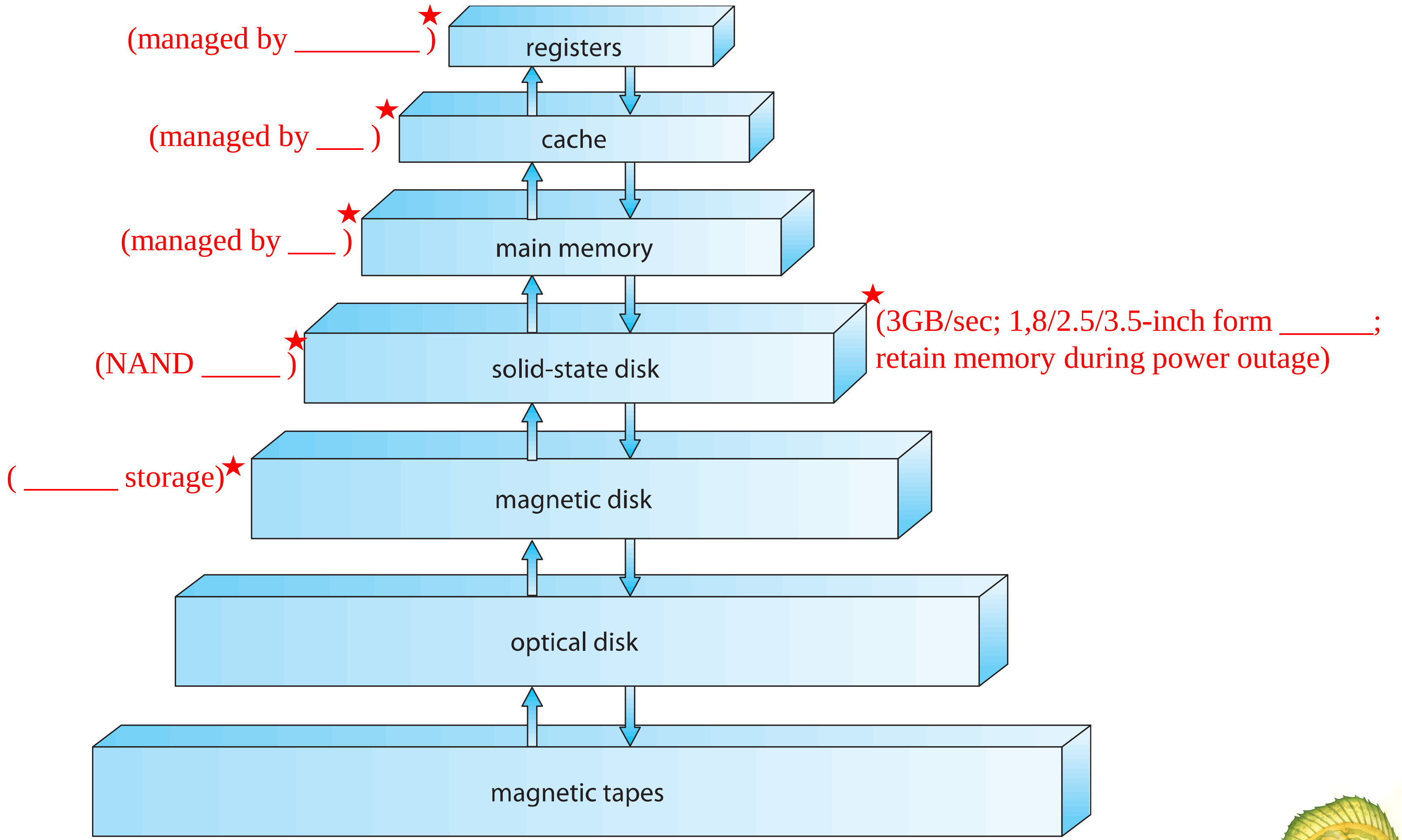
Caching – copying information into faster storage system; main memory can be viewed as a cache for secondary storage

★ (SW)

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

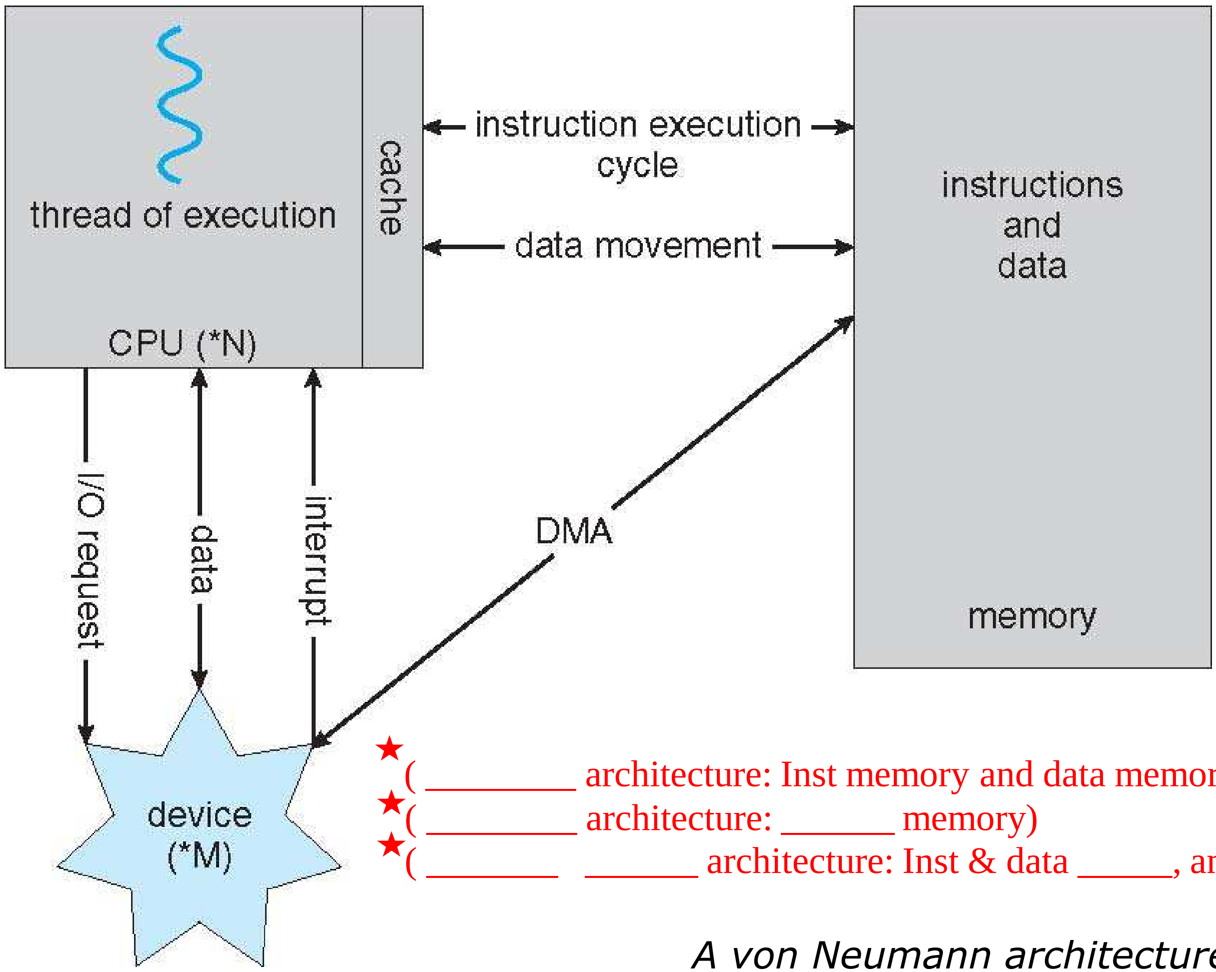
Cache smaller than storage being cached

- Cache management important design problem
- Cache size and replacement policy

Computer-System Architecture

- Most systems use a single general-purpose processor (PDAs through mainframes)
  - 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** ★ (Sun OS V4)
    2. **Symmetric Multiprocessing** ★ (Encore's Multimax, Sun OS V5 (Solaris 2))

How a Modern Computer Works

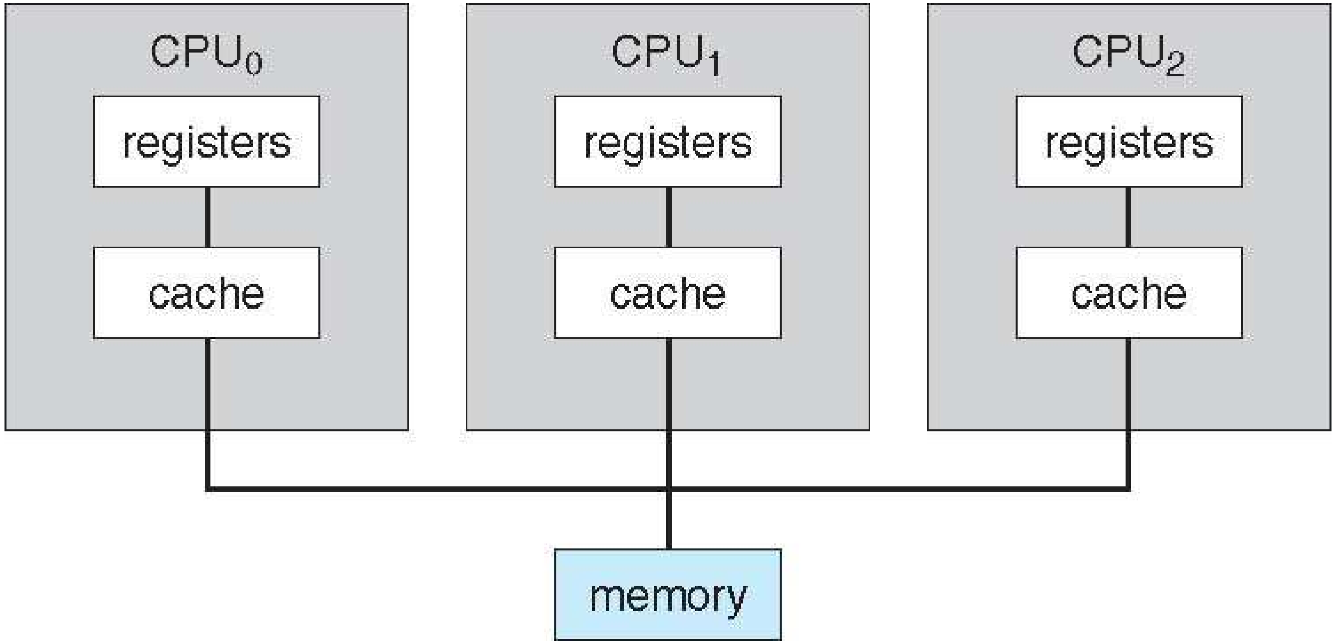


★(_________ architecture: Inst memory and data memory)
★(_________ architecture: _______ memory)
★(________ _______ architecture: Inst & data ______, and unified memory)

A von Neumann architecture

★ (Itanium 9500(*Poulson*): __-bit microprocessor, L2 512KB I, 256KB D per core, L3 32 MB, _ core, 2.53 GHz CPU clock, 12-wide issue architecture, multithreading, virtualization, die size ___ mm^2)

Symmetric Multiprocessing Architecture



★(iPhone ___: Nov. 2018; 64/256GB flash; __GB mem; iOS 12; 2.49 GHz 64-bit hexa-core Apple A12 Bionic SoC; GPU; ___ MP dual camera , iPhone 11 by Sept. 2019)

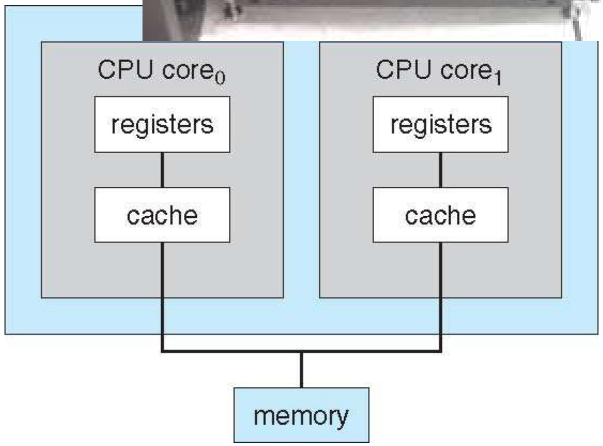
★(Galaxy S___: March 2019; 64/128/256GB flash; ___GB mem; _______ 9.0 Pie; SoC Samsung Exynos ____; 5G LTE-A; Removable storage microSD up to 400 GB)

A Dual-Core Design

★

- **UMA** and **NUMA** architecture variations
- Multi-chip and **multicore**
- Systems containing all chips vs. **blade servers**
  - Chassis containing multiple separate systems

★ (to minimize the use of physical space and ______ ; 180 servers/blade system)



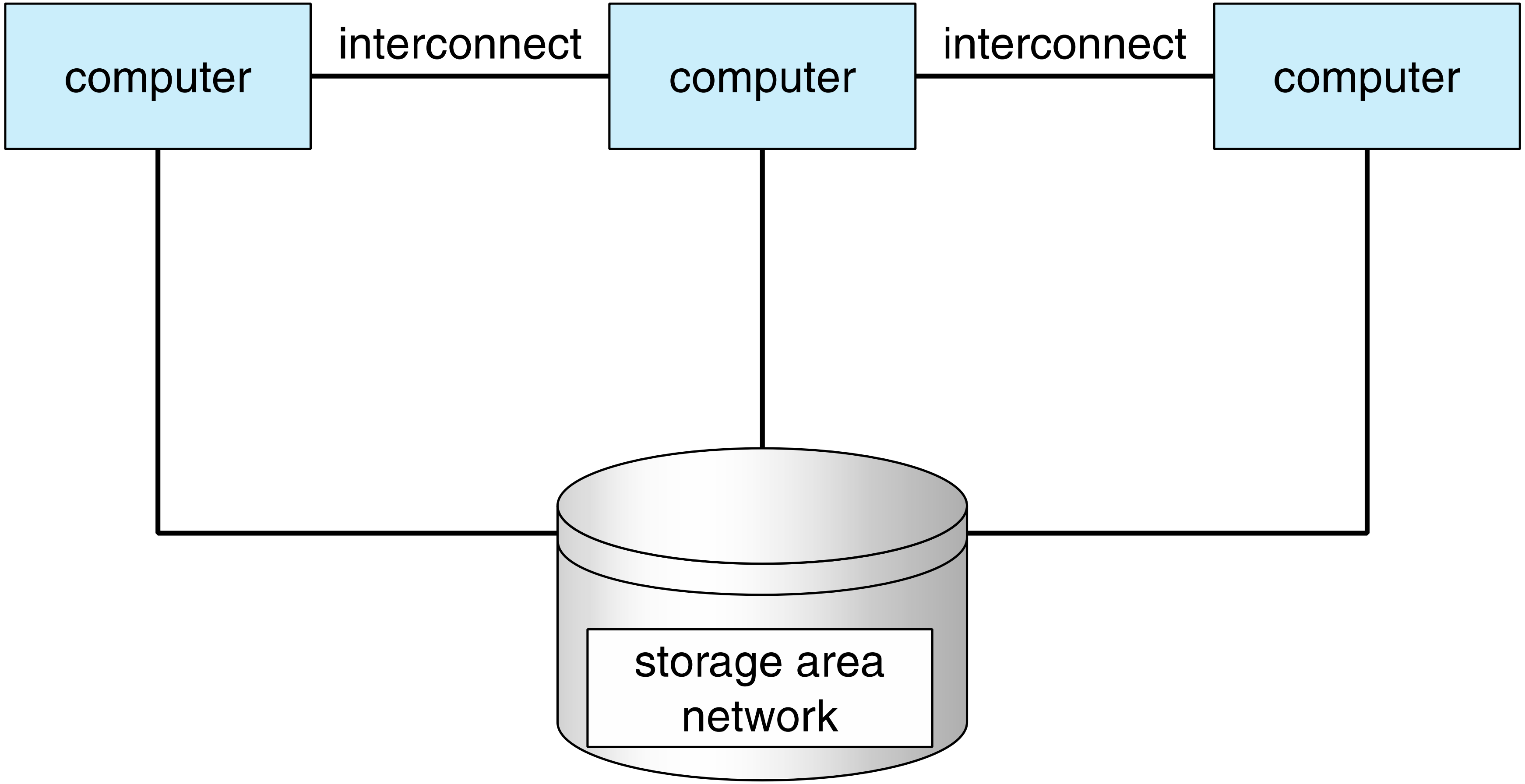
★ (ex) Intel Core 2 Duo E6750: each core has a CPU and L1 cache, one bus interface, ______ L2 cache; used for embedded, network, DSP , and graphics)

★ (Quad core/ hexa core)

Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a **storage-area network (SAN)**
    ★(provides access to consolidated, block level data storage)
  - 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**
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

Clustered Systems

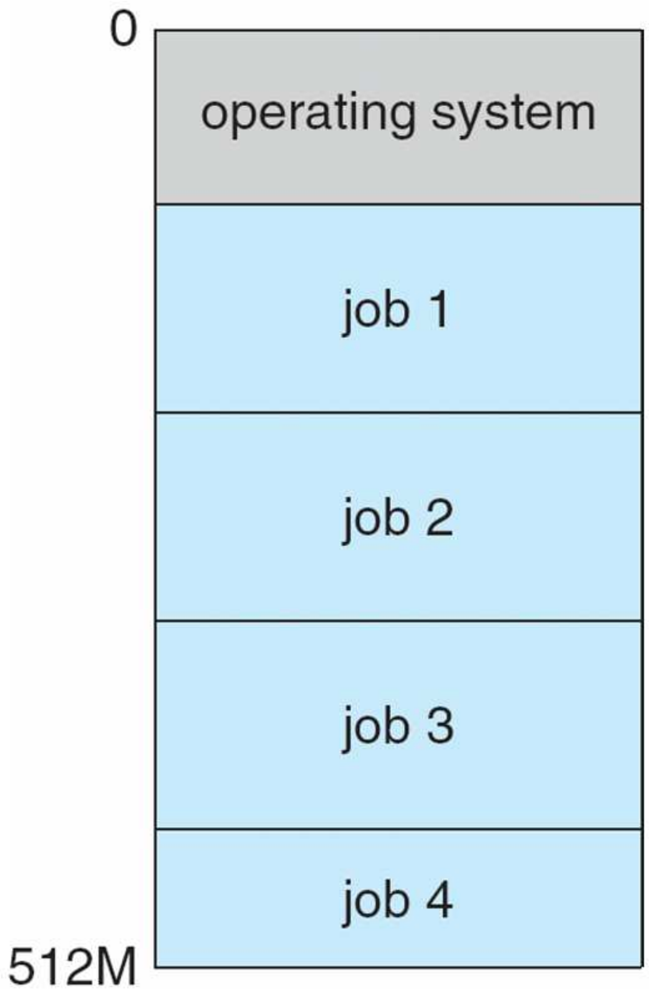


★ (Parallel cluster allows multiple hosts to access same data on the _______ storage such as Oracle Parallel Server; clustering over WAN)

Operating System Structure

- **Multiprogramming** needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so 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 it has to wait (for I/O for example), OS switches to another job
- **Timesharing** (**multitasking**) is logical extension in which CPU switches jobs so frequently that users can interact with each job while 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 run
  - **Virtual memory** allows execution of processes not completely in memory

Memory Layout for Multiprogrammed System



Operating-System Operations

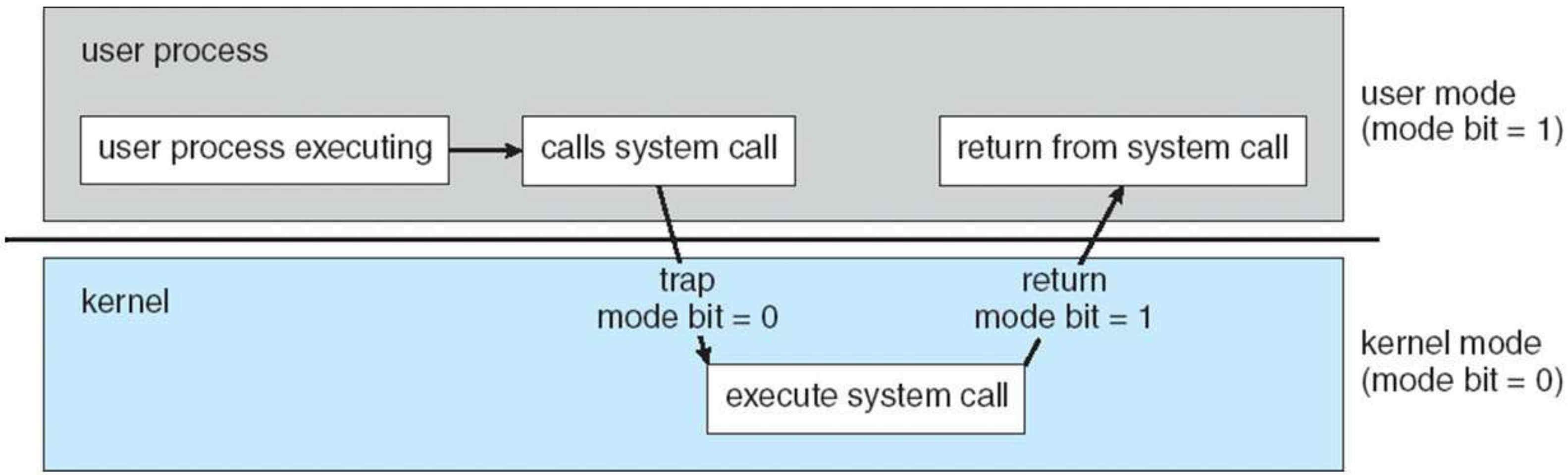
- **Interrupt driven** by hardware
- Software error or request creates **exception** or **trap**
  - Division by zero, request for operating system service
- Other process problems include infinite loop, processes modifying each other or the operating system
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware ★(there are instructions referring to this bit for ____ operation)
    - 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
- Increasingly CPUs support multi-mode operations
  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

★(the privilege is between kernel and user; Intel 64 family supports __ levels)

Transition from User to Kernel Mode

Timer to prevent infinite loop / process hogging resources

- Set to interrupt after specific period
- Operating system decrements counter
- 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

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 whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

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