CISC 7310X
C01b: Overview of Computer Systems and Operating Systems

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Acknowledgement

- This slides are a revision of the slides by the authors of the textbook

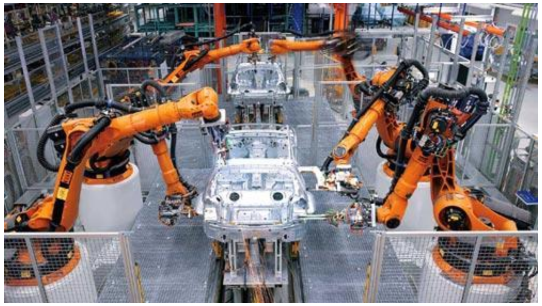
Outline

- Motivational discussion
- Definition of operating systems
- Computer organization overview
- Computer architecture overview
- Operating system structure
- Operating system resources management
- Protection and privacy
- Computing environments

Discussion Questions?

- Where may you find computers?
- Why do we need to study operating systems?

Various Computer Systems



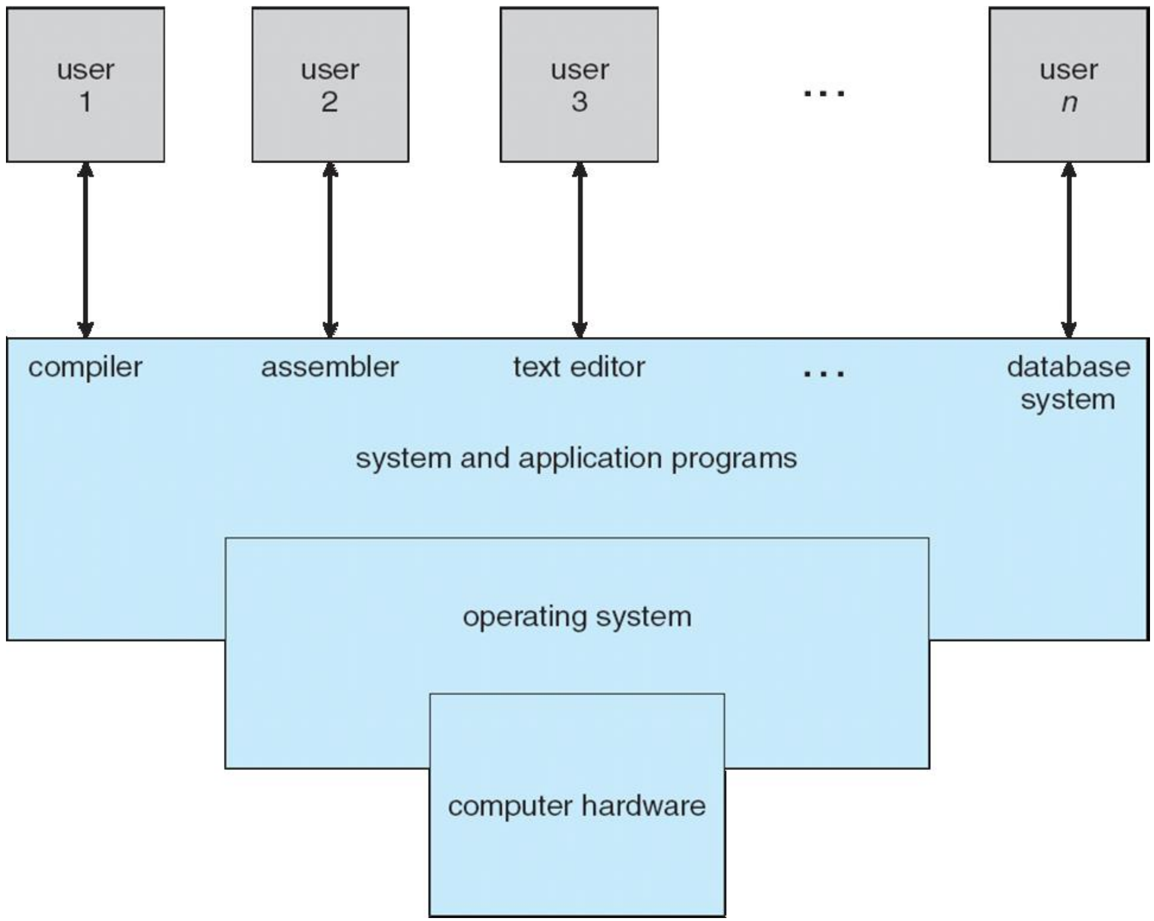
Motivations

- There are occasions we are better off to provide a system solution from ground-up.
  - Building a new system
  - Extending an existing system
- There are (more) occasions we can only provide a superior “system” solution when we have a good understanding how the system works.
- There are also occasions we can application the solutions (algorithms) in operating systems to application programs
- …

Discussion Questions

- So, lots of computers … What are in common and what are different? What is a computer system?

A Computer System: Four Components



Computer System Components

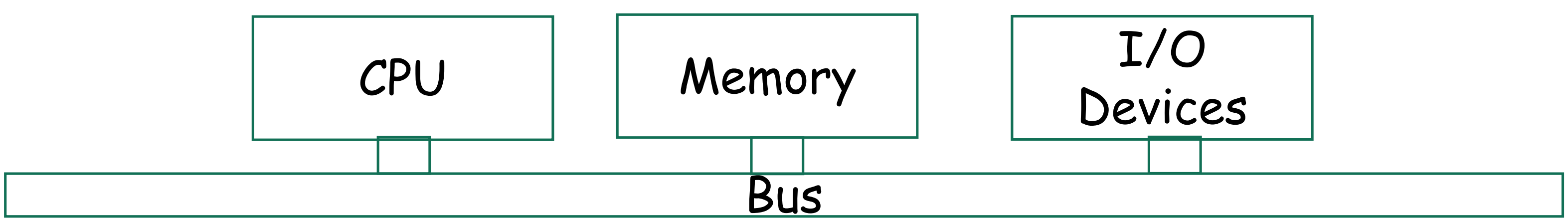
- Hardware: provides basic computing resources
  - CPU, memory, I/O devices
- Operating system
  - Controls and coordinates use of hardware among various applications and users
- Application programs: define the ways in which the system resources are used to solve the computing problems of the users
  - Word processors, compilers, web browsers, database systems, video games
- Users
  - People, machines, other computers

Questions?

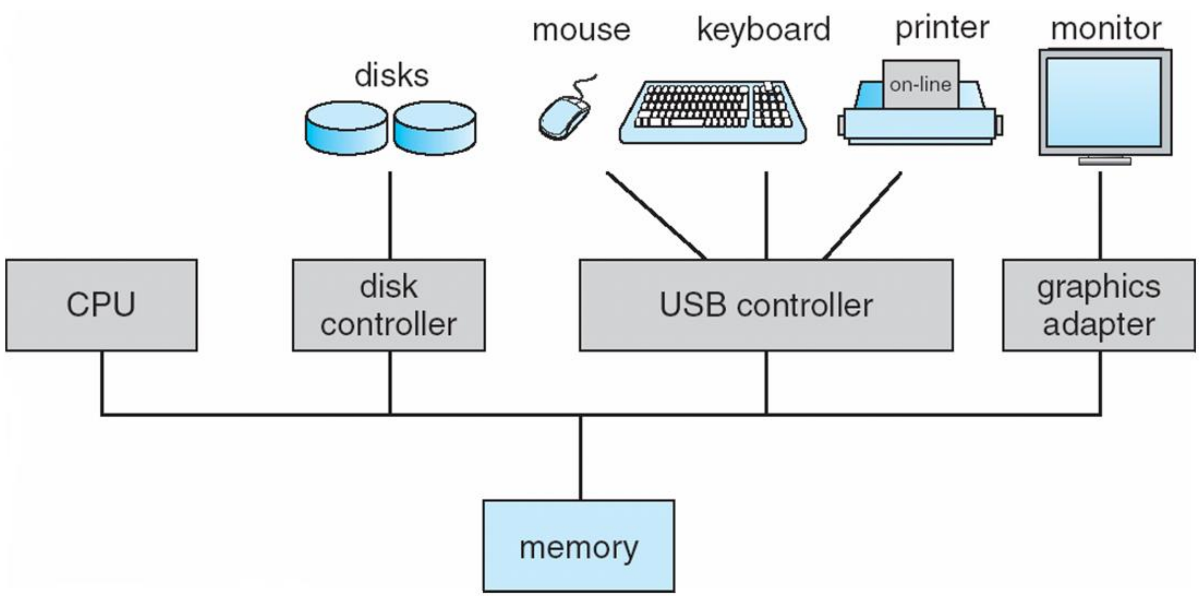
- What are major (top-level) computer system components?
- Does a user always mean a human user?

Major Hardware Components

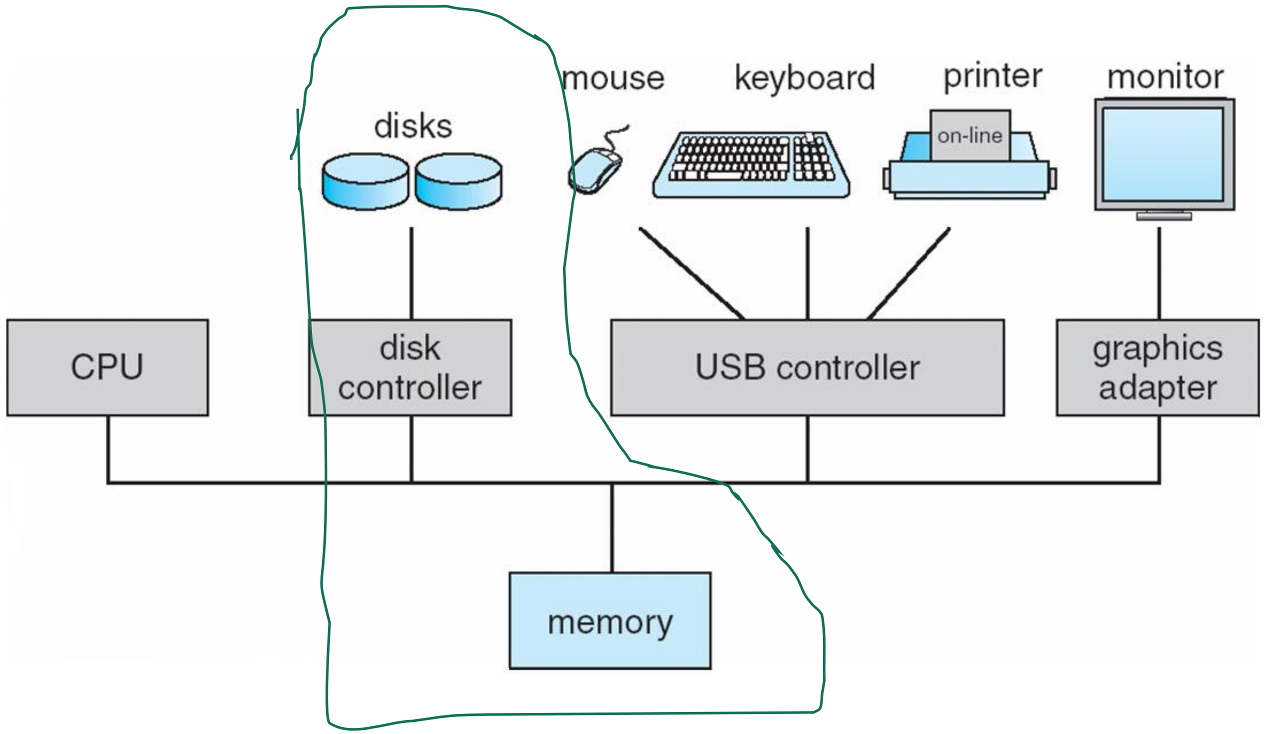
- Processors (CPU)
  - Multithreaded and multicore processors
- Main Memory (Memory)
- Secondary Memory (Disks)
- I/O Devices
- Buses



Major Hardware Component: An Example



Storage Structure



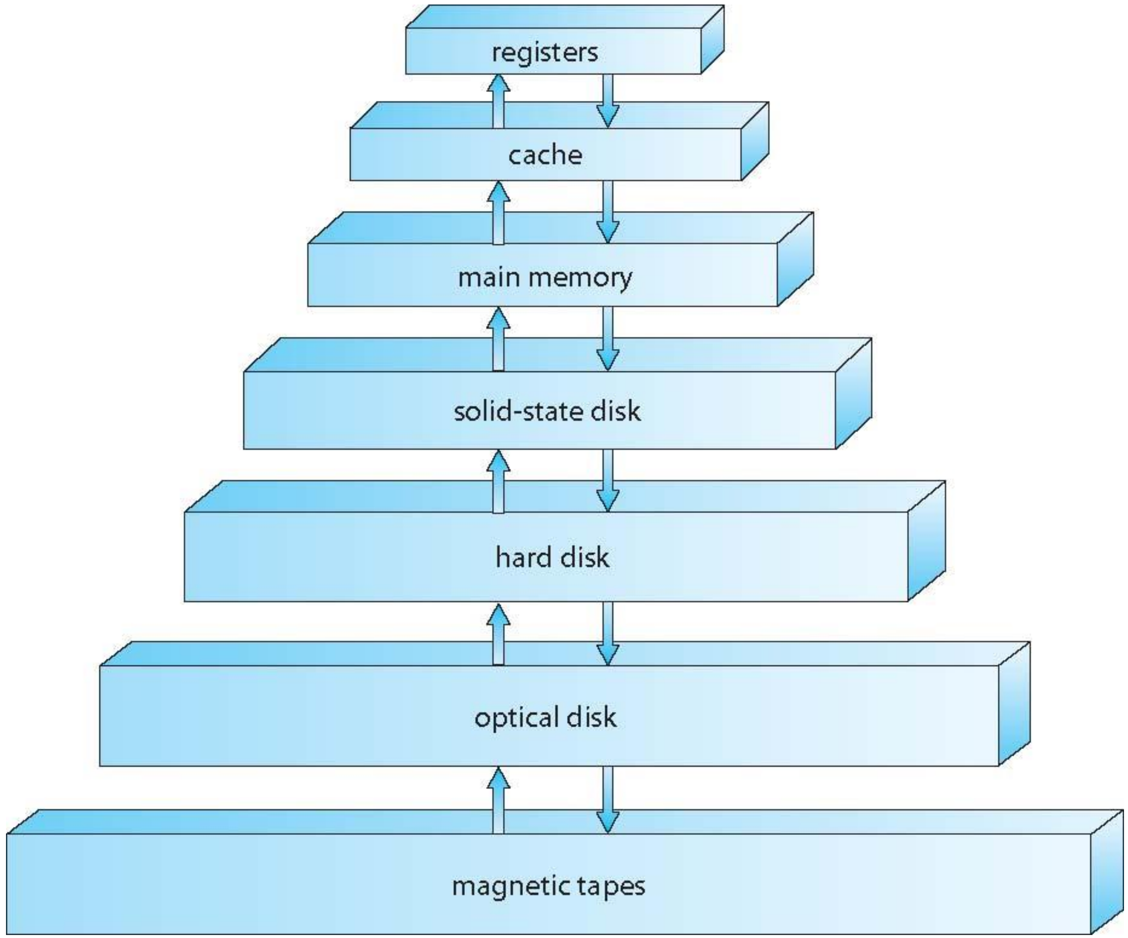
Main Memory and Secondary Storage

- 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
- Hard 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 hard disks, nonvolatile
  - Various technologies, 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
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

Storage-Device Hierarchy



Performance of Various Levels of Storage

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

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

Caching

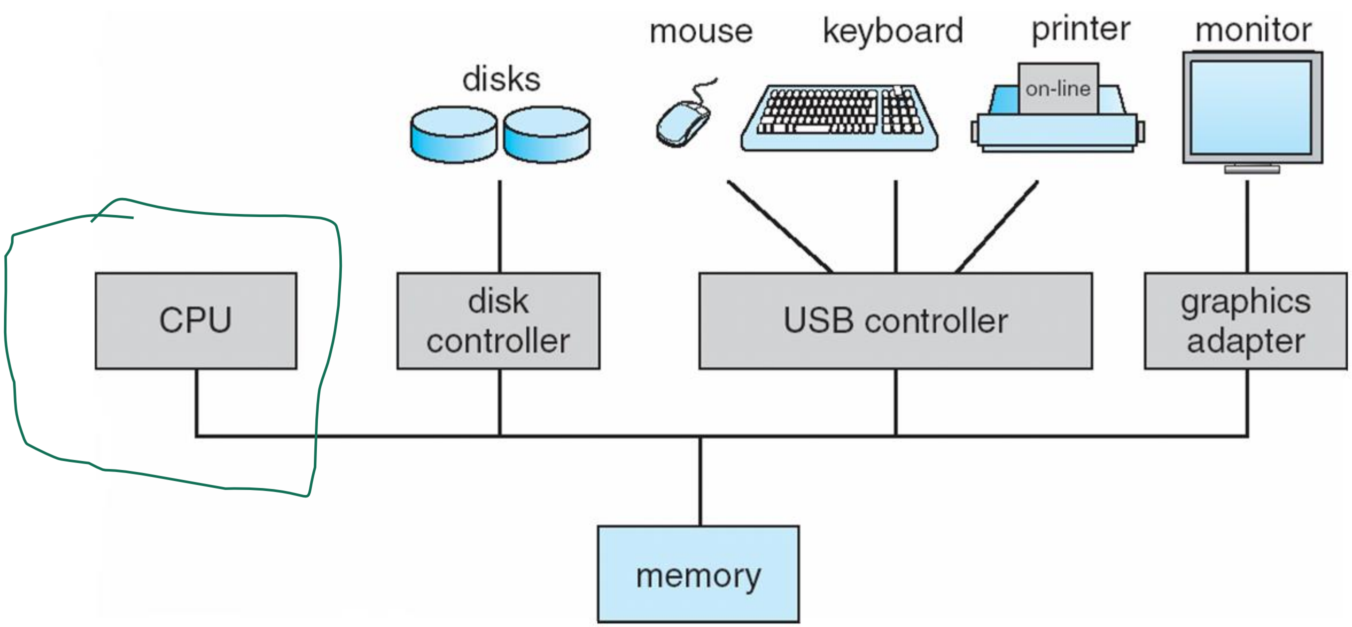
- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (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

Questions?

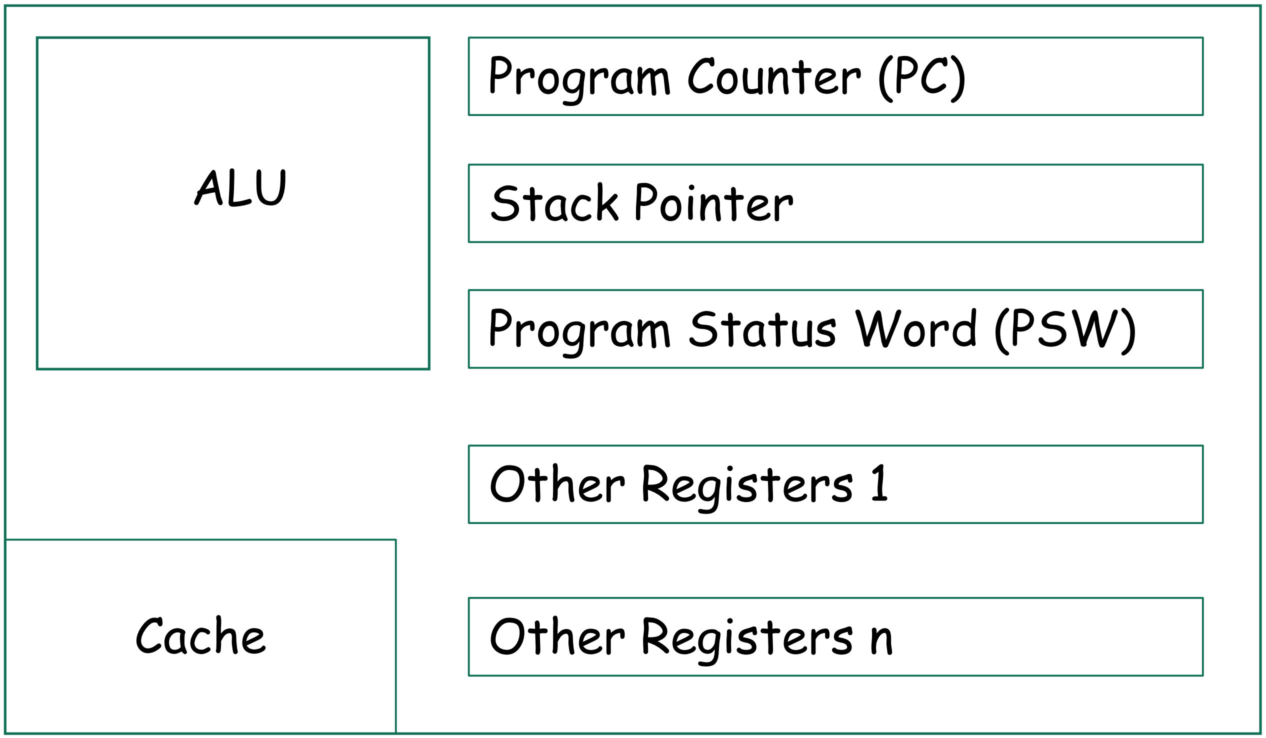
- Tiers of Storage
  - How is storage organized?
- Memory/Storage hierarchy
  - How does storage-device hierarchy look like? Why is storage organized in this fashion?
- Caching
  - What is caching?

Processor

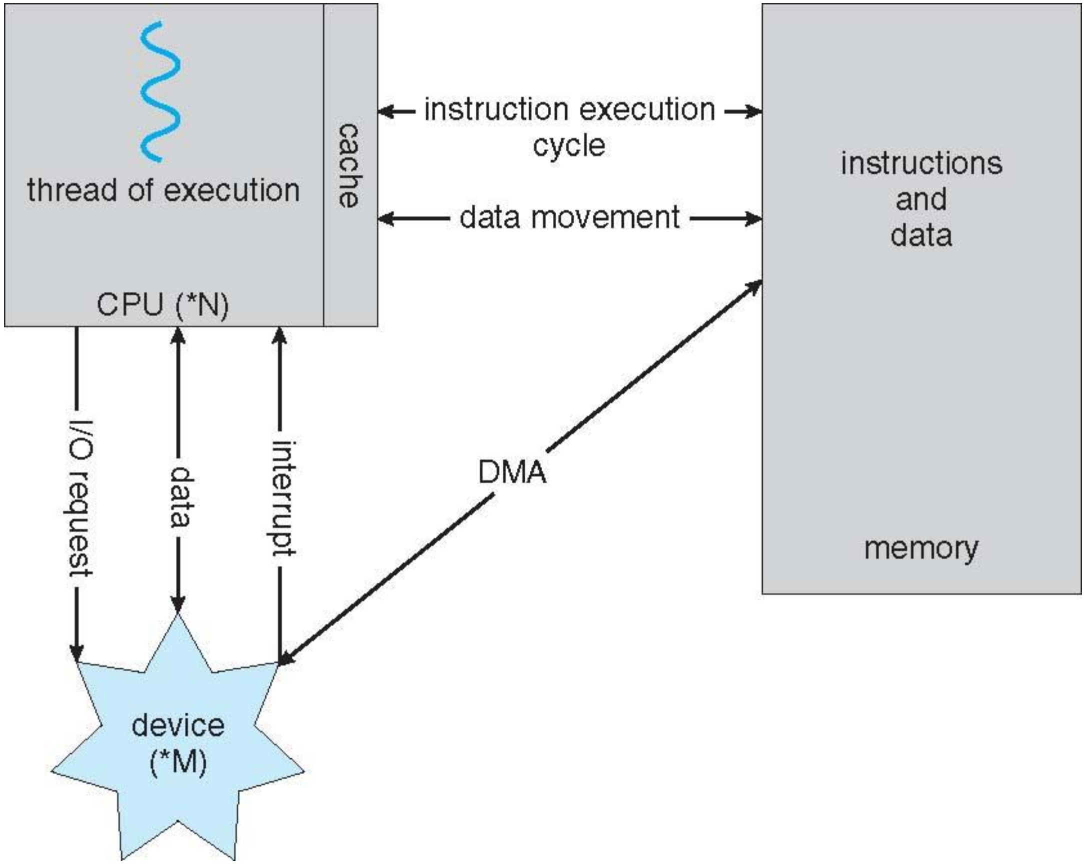
- CPU = Central Processor Unit



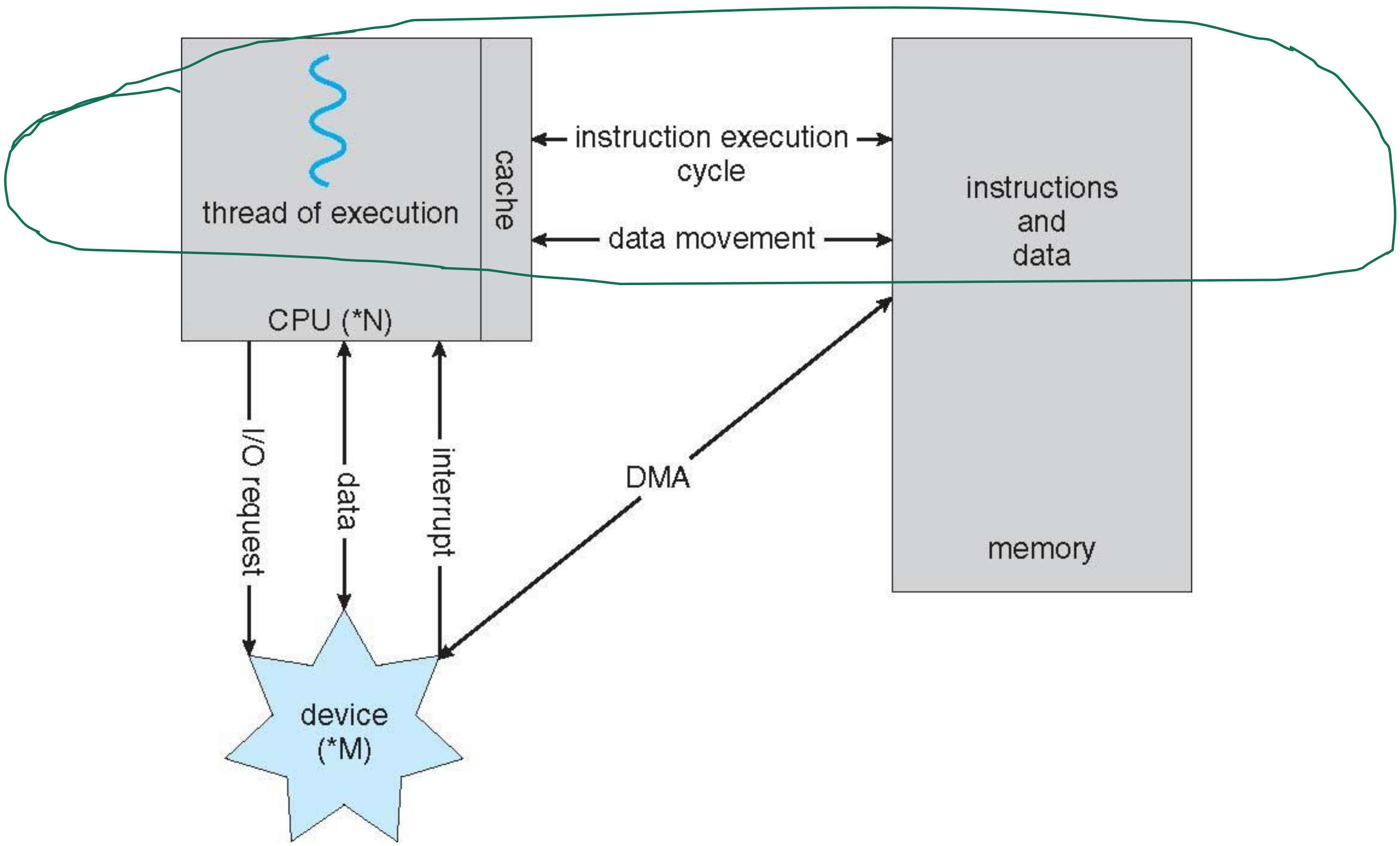
Processors


ALU
Program Counter (PC)
Stack Pointer
Program Status Word (PSW)
Other Registers 1
Cache
Other Registers n

A von Neumann Architecture



A von Neumann Architecture



Instructions

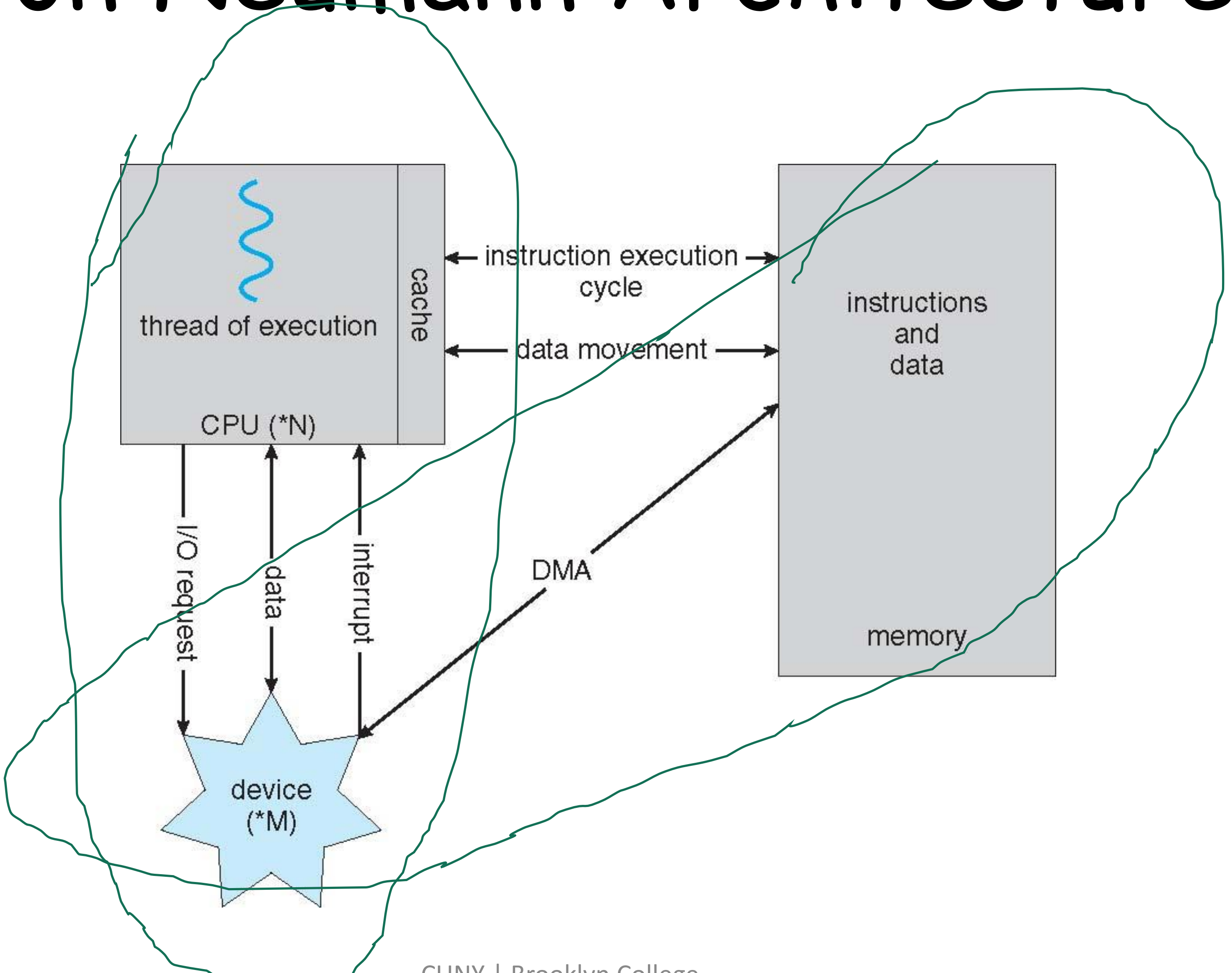
- Basic cycle
  - Fetch, decode, execute
  - Enhance: e.g., pipelining
- Instruction set
  - Examples
    - x86 (i386 for 32-bit; amd64 for 64-bit)
    - ARM

Fetch	Decode	Execute	
	Fetch	Decode	Execute

Questions?

- Processor
- Instruction set and cycle

A von Neumann Architecture



Input and Output (I/O)

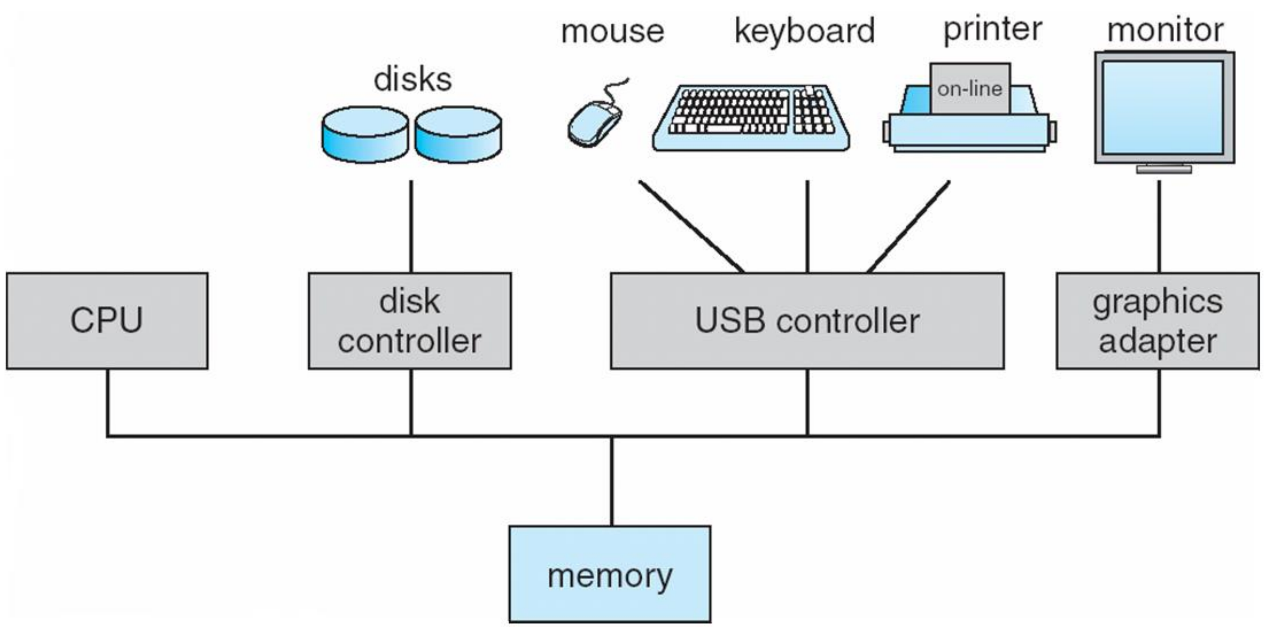
- Busy waiting
- Interrupted I/O
- Direct memory access (DMA)
- Discuss more next week

Questions?

- Concept of I/O and different types of I/O mechanisms

This is an Example of a Simple System

- Many systems use a single general-purpose processor, but …



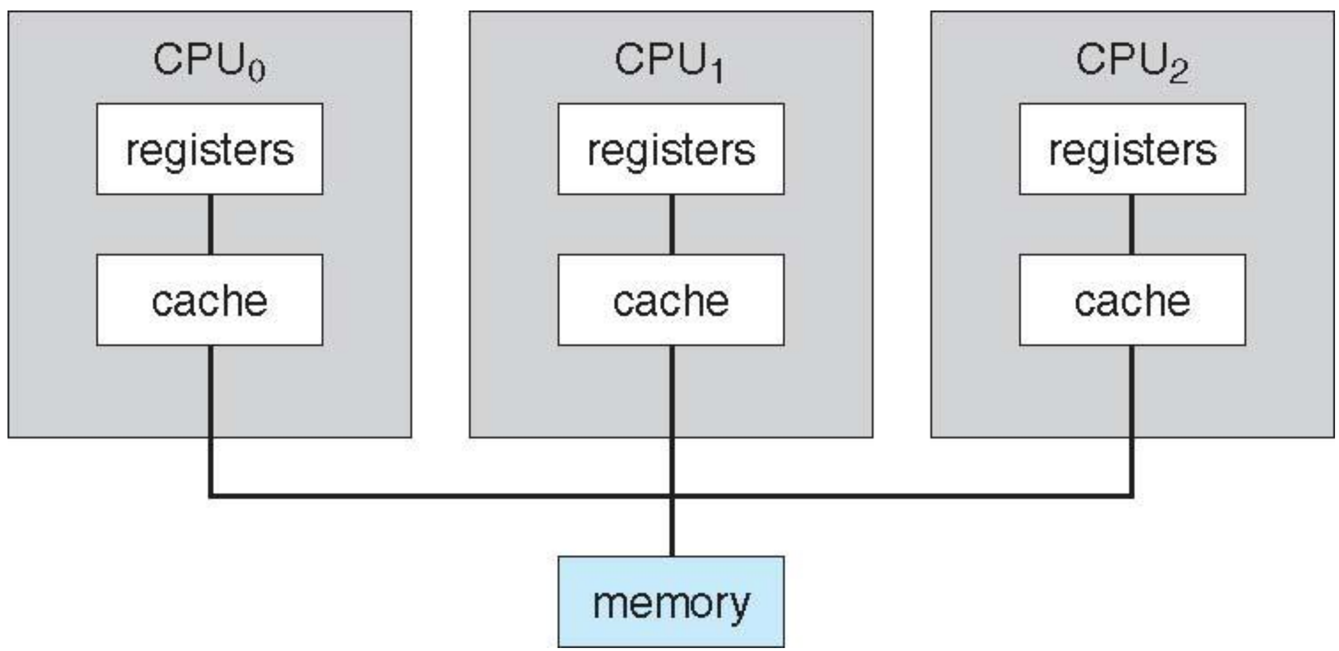
Computer System Architecture

- Many systems use a single general-purpose processor
  - Many 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** – each processor performs all tasks

Architecture: Examples

- Symmetric multiprocessing
  - Multi-core design
- Clustered systems

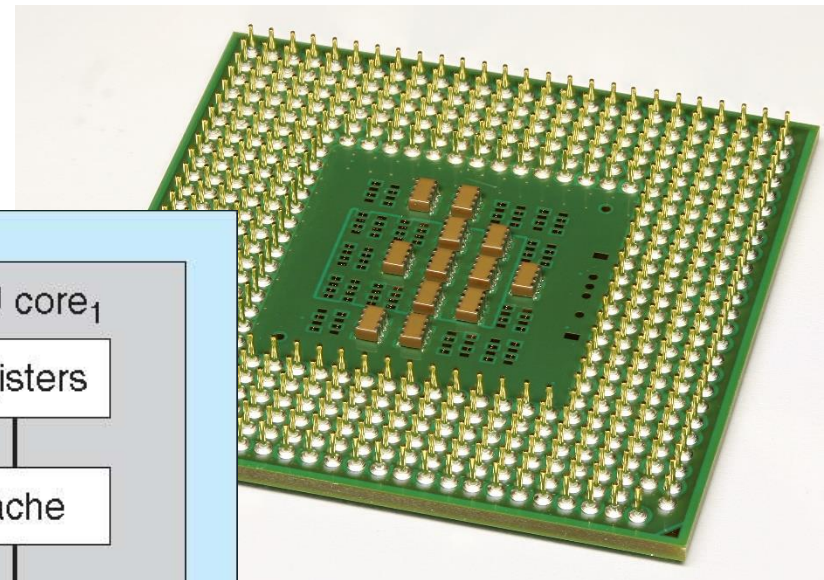
Symmetric Multiprocessing Architecture

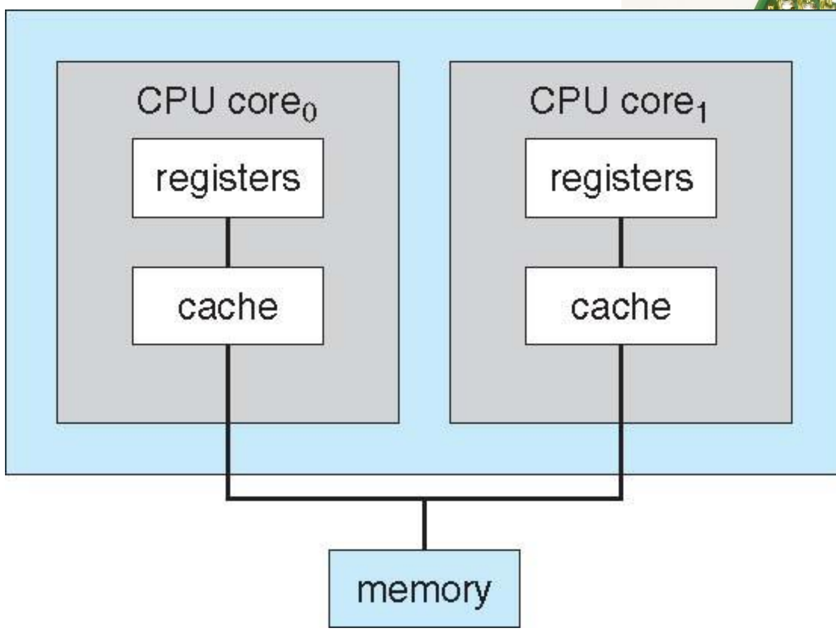


Multi-chip and Multicore

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

A Dual-Core Design

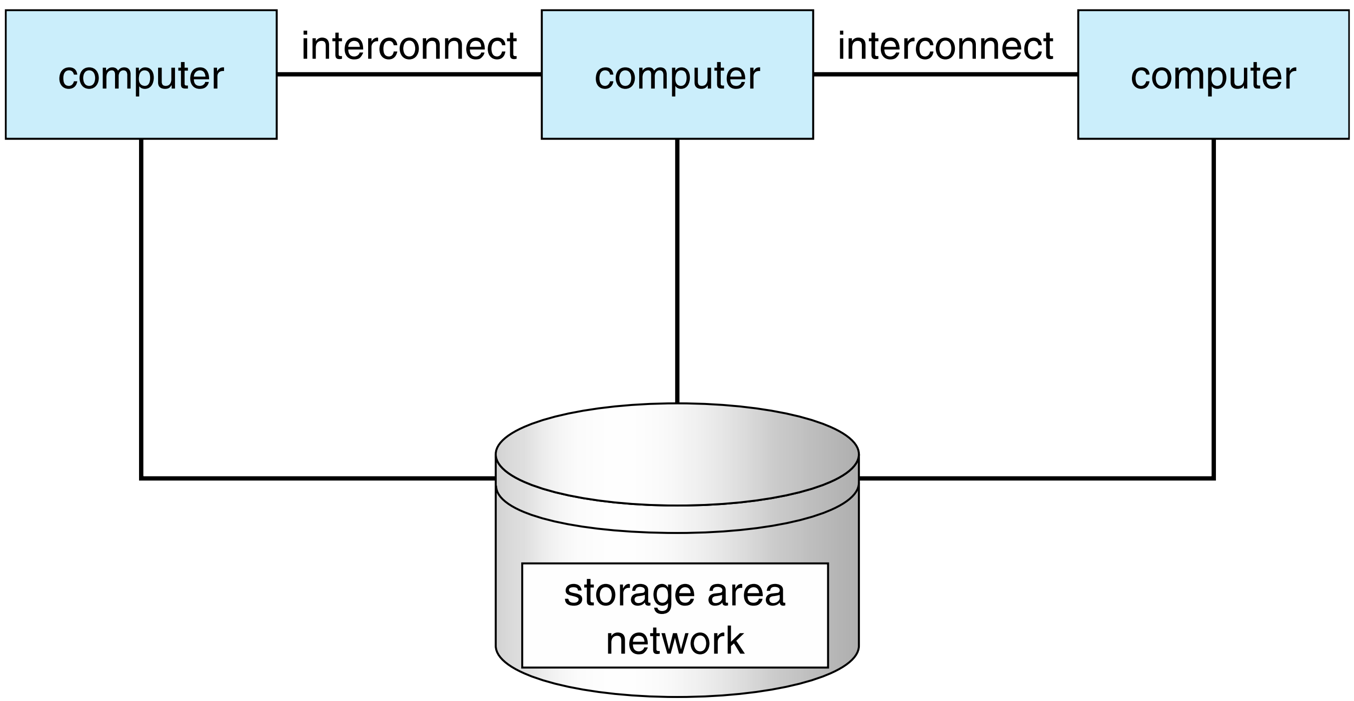




Clustered Systems

- 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** 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: Example



Questions?

- How does a typical von Neumann architecture look like?
- Types of architecture

Discussion Question?

- What is an Operating System?

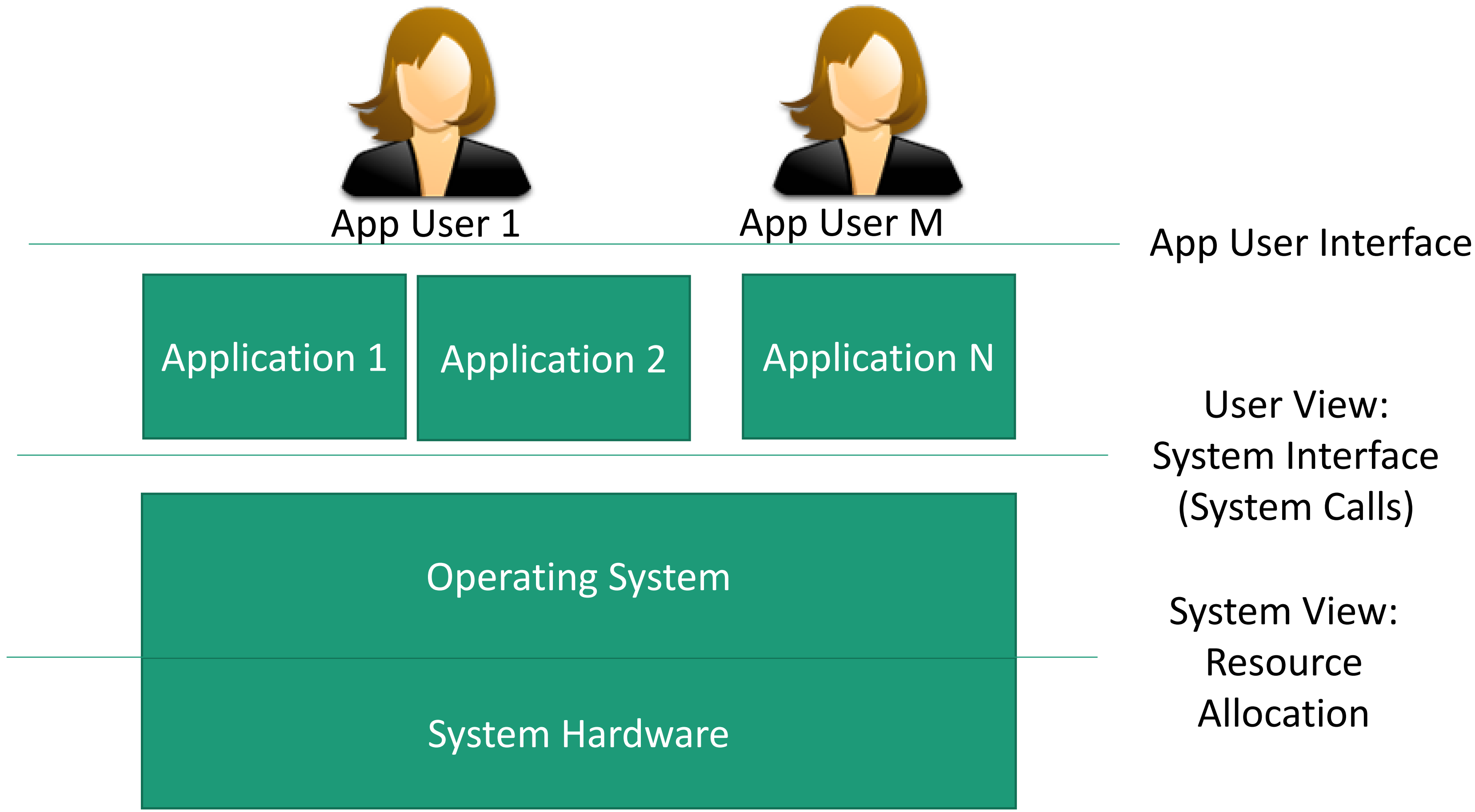
Operating System Definition: Two Roles

- An OS controls and coordinates use of hardware among various applications and users, and servers two major roles
- As a resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- As a control program
  - Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition: Two Views

- A large piece of software function as
  - an extended machine (user view)
    - to provide an “beautiful” interface for application programs for application developers
  - a resource manager (system view)
    - to provide a “beautiful” allocation scheme to share the processors, memories, and I/O devices in a “computer system”

Operating System



Did you say, “beautiful”?

- Question & Discussion: what is “beautiful”?

But, No Universally Accepted Definition

- "Everything a vendor ships when you order an operating system" is a 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.

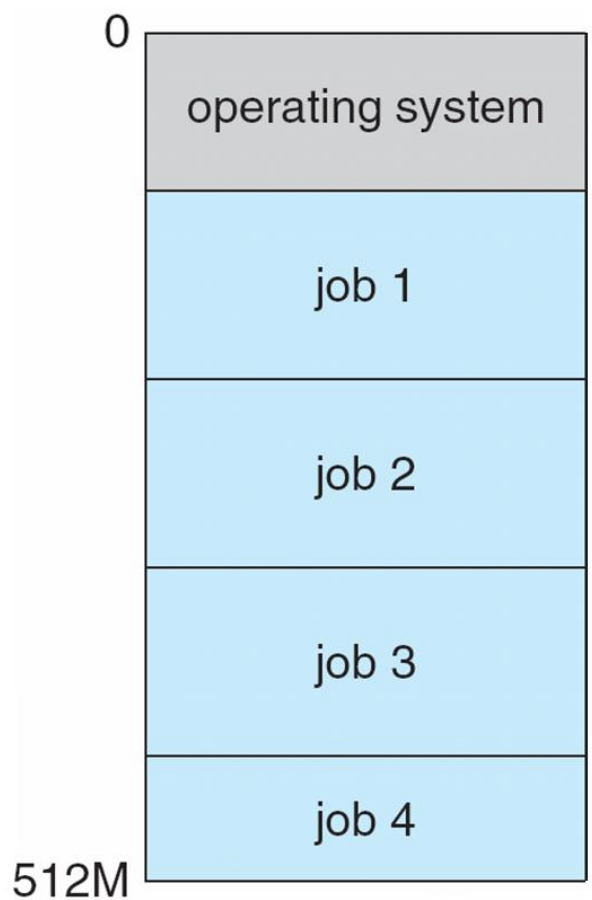
Questions

- Concept of operating systems
  - Two roles
  - Two views
- Is there a universally accepted definition? But, what is universally accepted?

Operating System Structure

- **Multiprogramming** (**Batch system**) 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



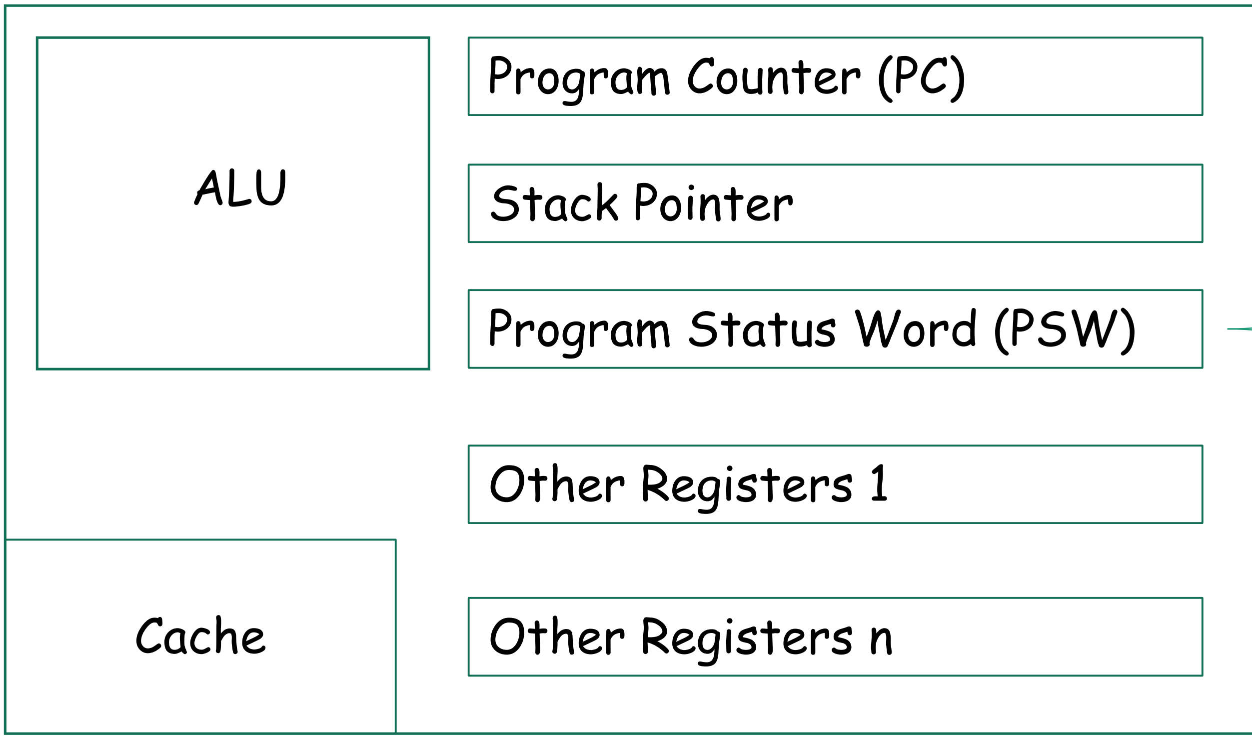
Operating-System Operations

- **Interrupt driven** (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap):**
    - Software error (e.g., division by zero)
    - Request for operating system service
    - Other process problems include infinite loop, processes modifying each other or the operating system

Operating-System Operations

- **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
- Increasingly CPUs support multi-mode operations
  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Processors



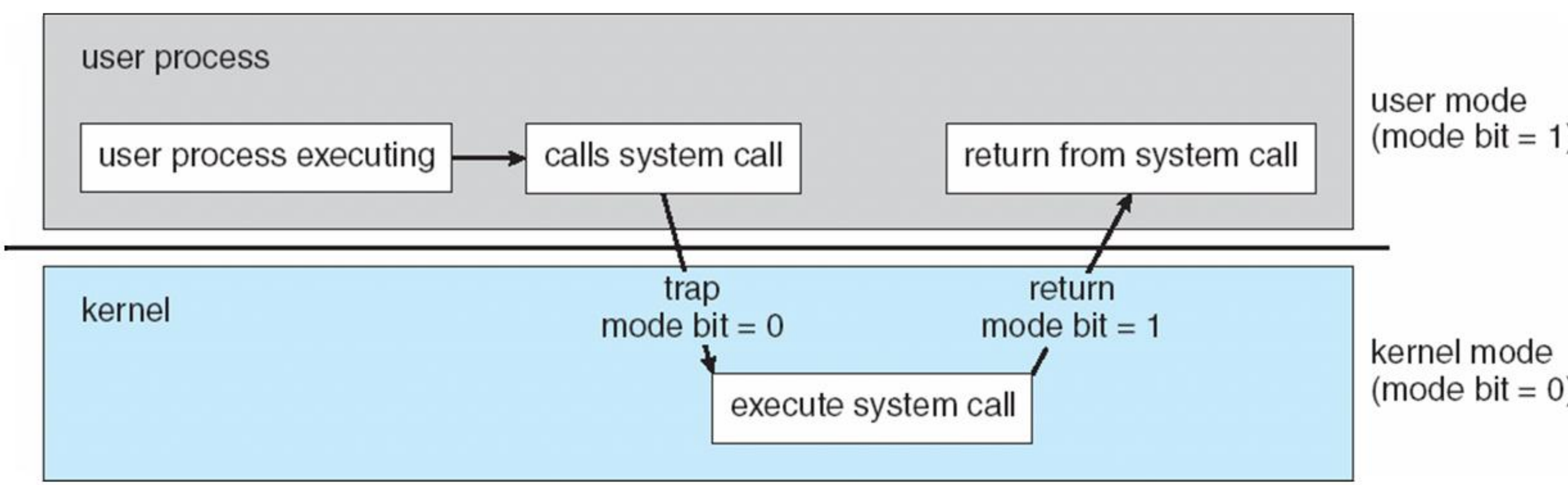
Kernel mode: can execute all instructions and access all hardware features

User mode: can execute subset of instructions and access subset of hardware features

Transition from User to Kernel Mode

- Timer to prevent infinite loop / 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

Transition from User to Kernel Mode



Questions?

- Operating system structure
- Dual mode (kernel and user mode)
- Transition from use to kernel mode

Operating System Resource Management

- Process management
- Memory management
- Storage management
- Mass-storage (disk) management
- I/O subsystem
- Protection and privacy

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

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- 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 deallocating memory space as needed

Storage Management

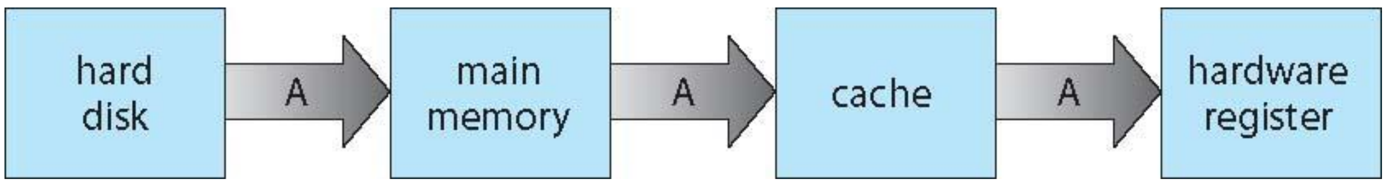
- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential 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 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
  - 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)

Migration of Data 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
  - Various solutions covered in Chapter 17

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

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

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 defined and controls managed, then also associated with each process, file
    - **Privilege escalation** allows user to change to effective ID with more rights

Questions?

- How does an OS manage the computer resources?
  - Process management
  - Memory management
  - Storage management
  - Mass-storage (disk) management
  - I/O subsystem
- Protection and privacy
- How is an operating system loaded?