OPERATING SYSTEMS

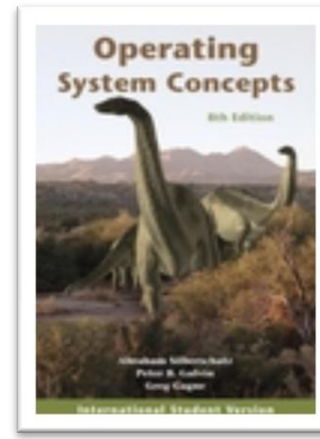
CS302

OUTLINE

- Introduction
- Processes
- Threads
- CPU Scheduling
- Process Synchronization
- Deadlocks
- Memory Management
- Virtual Memory
- File System Interface
- File System Implementation
- I/O Systems
- Mass Storage Structure

Primary Text

- "OPERATING SYSTEM CONCEPTS ", by Abraham Silberschatz, Peter B.Galven and Greg Gagne, The 8th Edition, Wiley & Sons, 2010.



Optional:

- "OPERATING SYSTEMS: A DESIGN-ORIENTED APPROACH"" by Charles Crowley, McGraw-Hill, ISBN:0256151512
- "OPERATING SYSTEMS DESIGN AND IMPLEMENTATION" by Tanenbaum and A.Woodhull, Prentice Hall Software Series, 2006.

Grading Policy

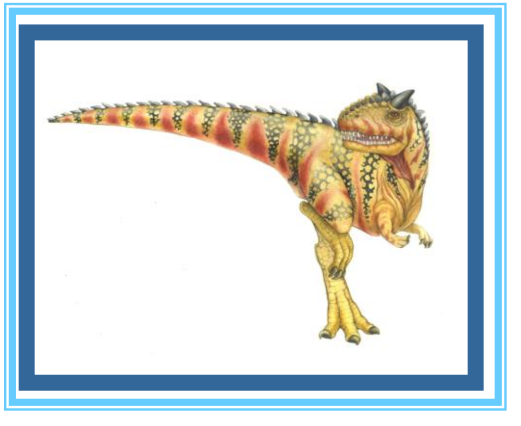
- **Midterm Exams**
  - **First Midterm** 11/3/2012 10%
  - **Second Midterm** 22/4/2012 15%
- **Quizzes** 5%
  - **Quiz 1** Week 4 18/02/2012
  - **Quiz 2** Week 14 05/05/2012
- **Homework's** 5%
- **Project** 5%
- **Final Lab** 20%
- **Final Exam** 40%

NOTE: NO REPLACEMENT EXAMS

Web Site

- www.cs302.yolasite.com

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
  - File-System Management
  - Mass-Storage Management
  - I/O Systems

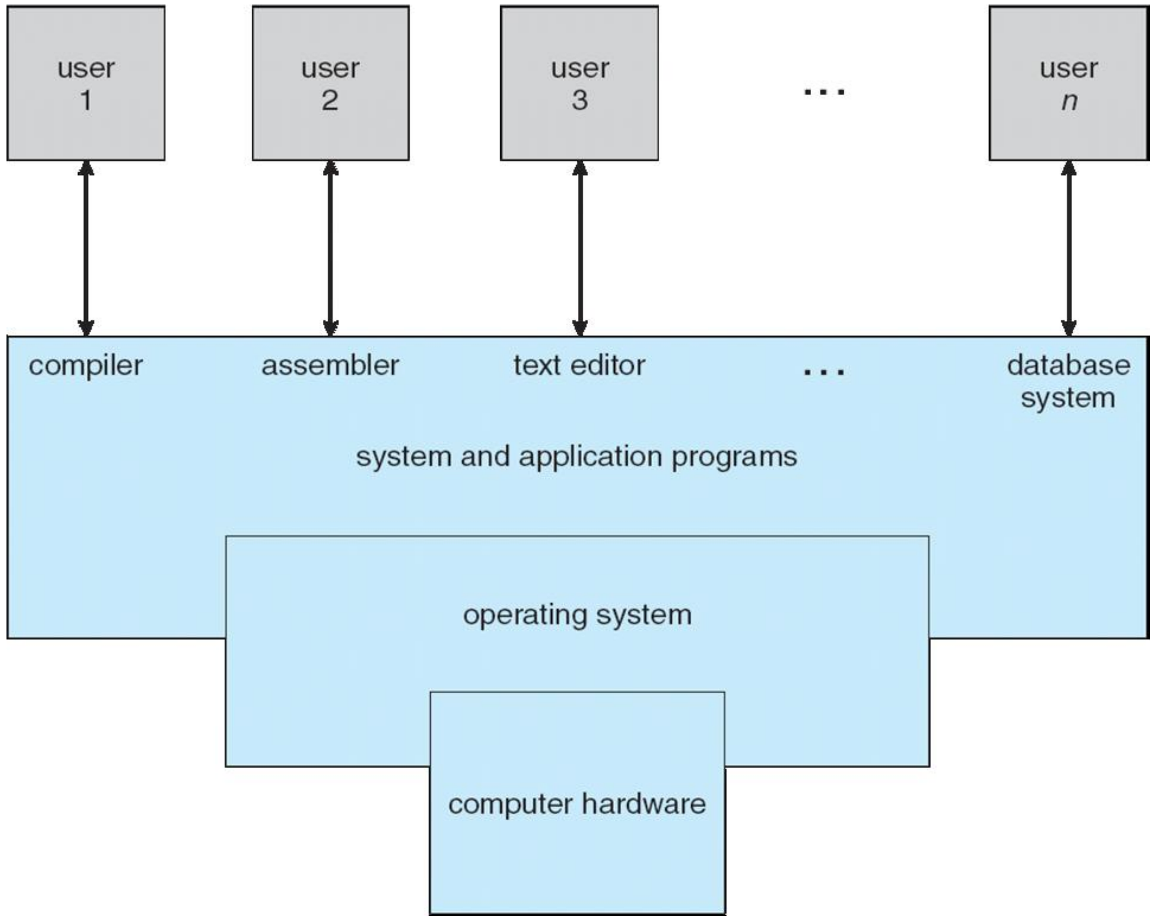
What is an Operating System?

- 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



Operating System Definition

- "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.
- 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

An operating system is a software that manages the computer hardware, as well as providing an environment for application programs to run.

Computer Startup

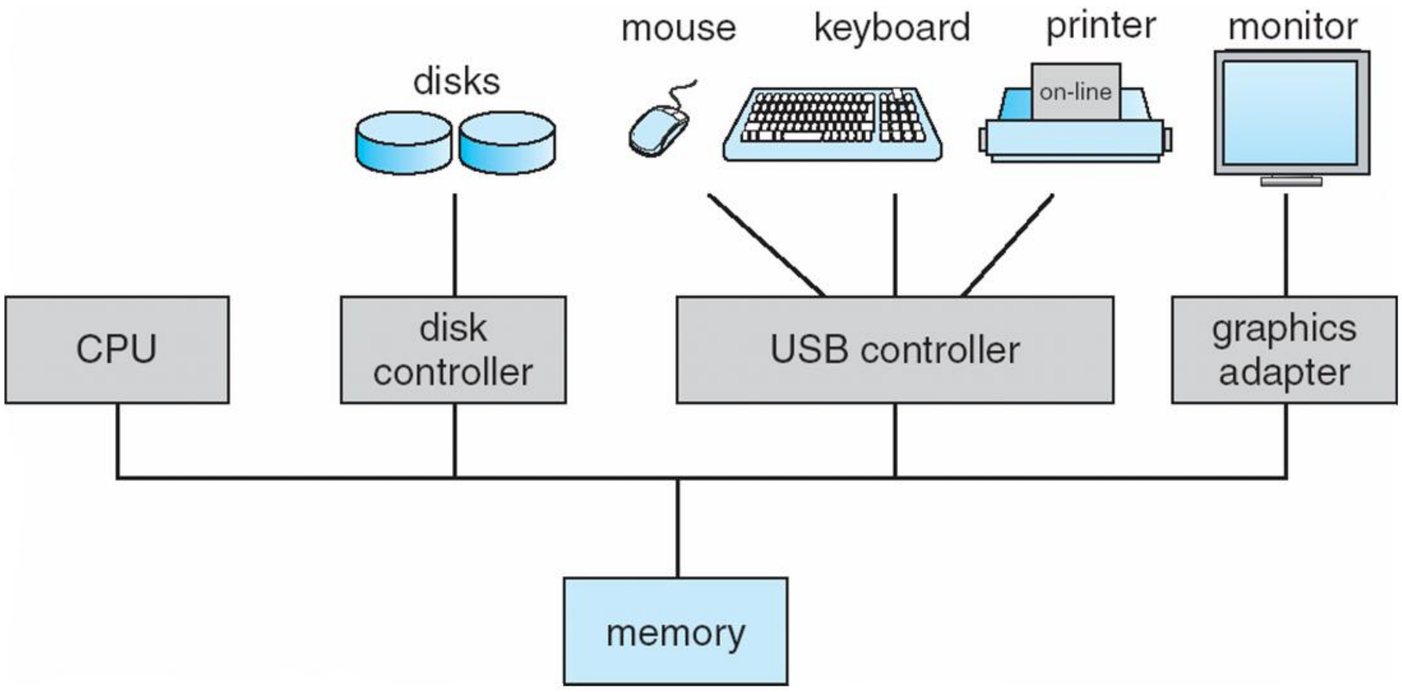
- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EEPROM (**E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory), generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Computer-System Organization

- Computer-System Operation
- Storage Structure
- I/O Structure

Computer System operation

- 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 (cont.)

Interrupt:

- The occurrence of an event is usually signaled by an interrupt from either the hardware or the software.
  - Hardware interrupts by sending a signal to the CPU through system bus.
  - Software interrupts by executing a special operation called a **system call.**

Computer System Operation (cont.)

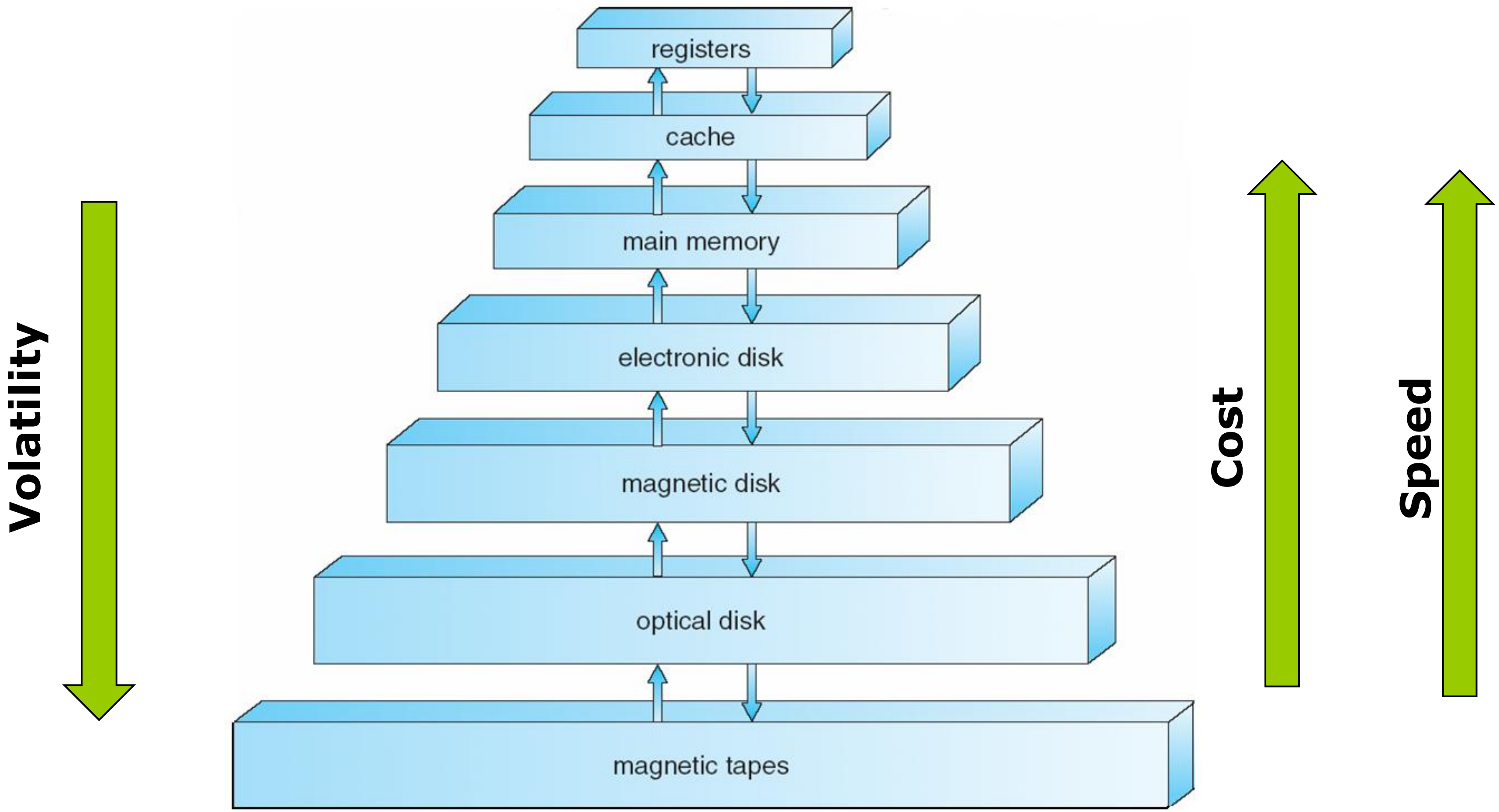
Common Functions of Interrupts

- 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
- 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
- An operating system is **interrupt driven**

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 – 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

Storage-Device Hierarchy



I/O Structure

- each device controller is in charge of a specific type of device.
- A device controller maintains some *local buffer* storage and a set of *special-purpose registers*.
- OS have a **device driver** for each device controller. This device driver understands the device controller and presents a uniform interface to device to the rest of the operating system.

I/O Structure(cont.)

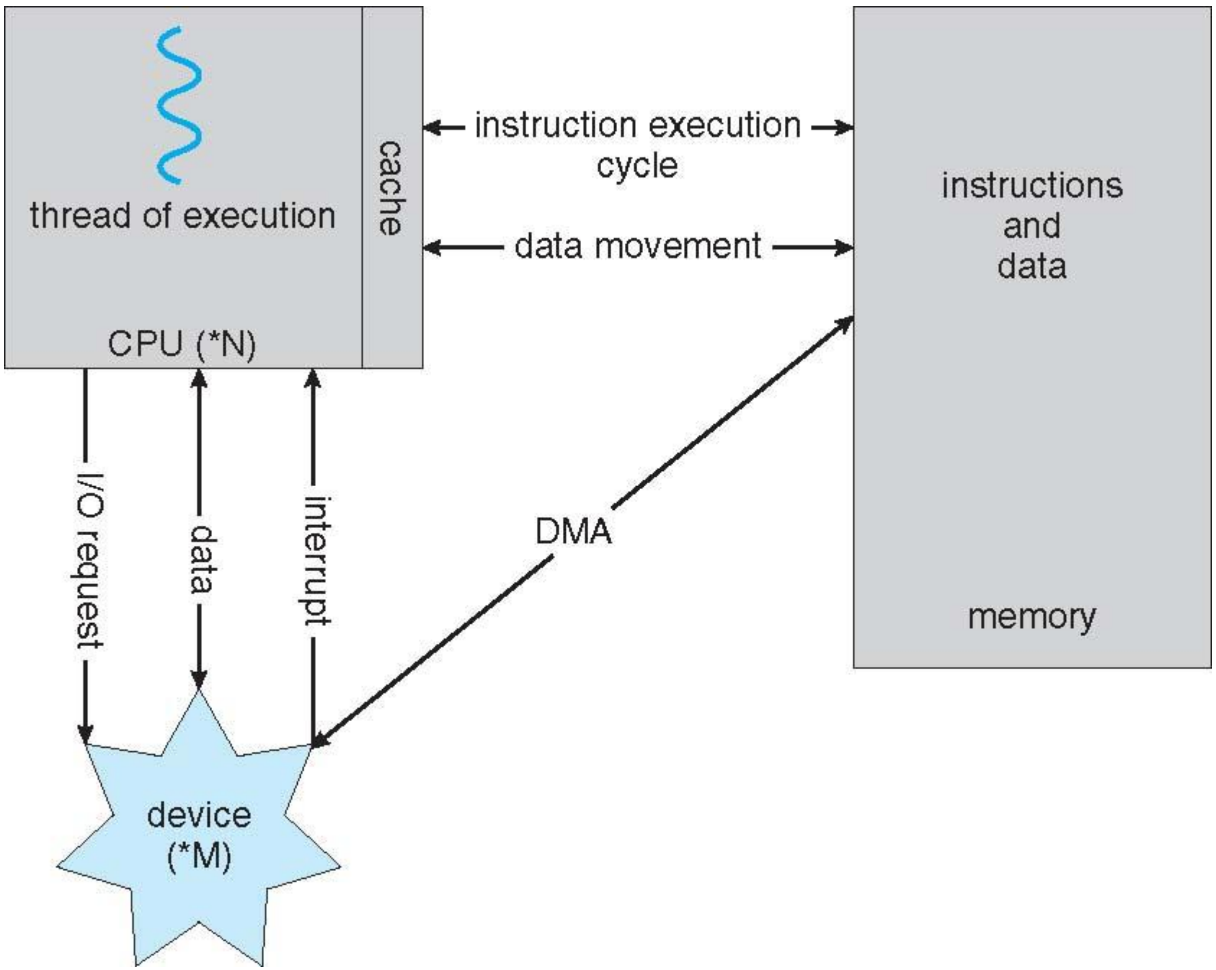
- To start ***interrupt-driven I/O operation***,
  1. The device driver loads the appropriate registers within the device controller.
  2. The device controller examines the contents of registers to determine what action to take.
  3. The controller starts the transfer of data from the device to its local buffer.
  4. Once the transfer of data is complete, the device controller informs the device driver via an interrupt that it has finished its operation.
  5. The device driver returns control to the OS.

What is the problem of this form?

I/O Structure (cont.)

- ***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

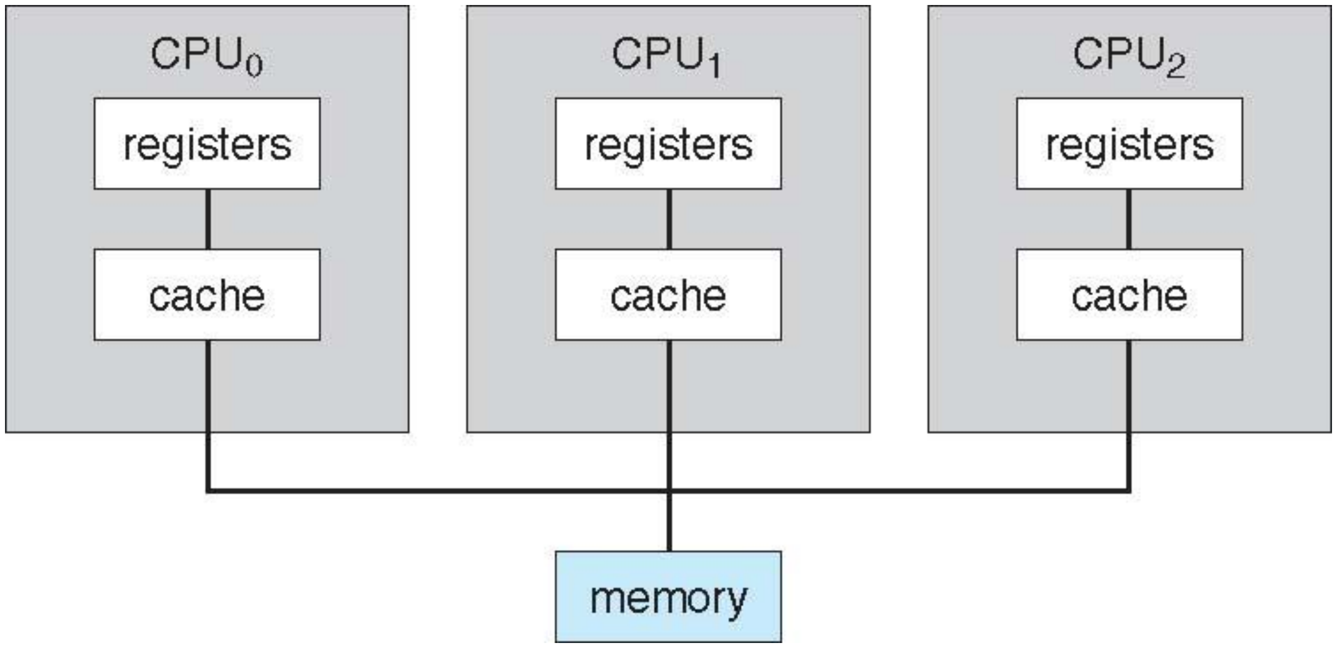
How a Modern Computer Works



Computer-System Architecture

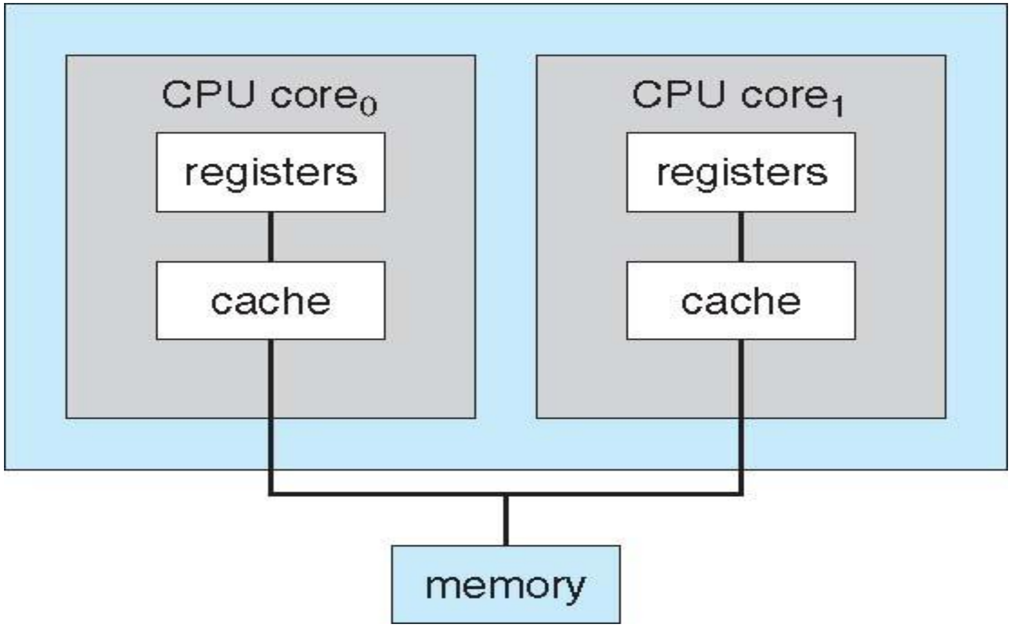
- Most systems use a single general-purpose processor (PDAs through mainframes)
  - Most systems have special-purpose processors as well
- Multiprocessors systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems
  - Advantages include
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – (fault tolerance)
  - Two types
    1. Asymmetric Multiprocessing (master-slave scheme)
    2. Symmetric Multiprocessing (peer-to-peer scheme)

Symmetric Multiprocessing Architecture



A Dual-Core Design

- Processors were originally developed with only one core.
- The core is the part of the processor that actually performs the reading and executing of the instruction. Single-core processors can only process one instruction at a time
- Dual-core processor contains two cores (Such as Intel Core Duo).



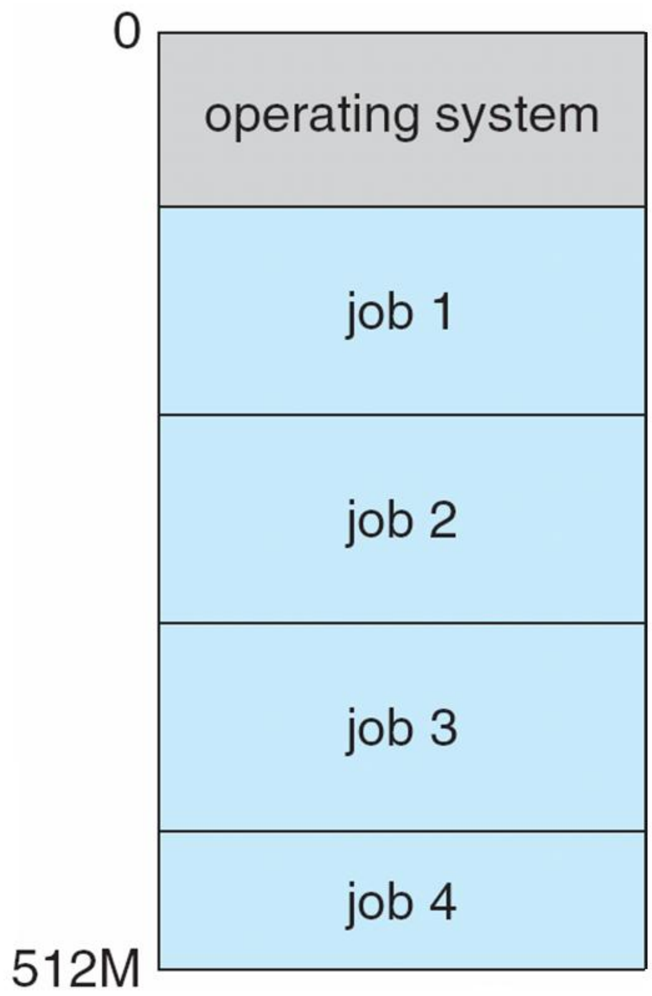
Operating System Structure

- ***Multiprogramming*** needed for efficiency
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute. (Increase CPU Utilization.)
  - 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

Operating System Structure (cont.)

- ***Timesharing (multitasking)*** is logical extension of multiprogramming 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

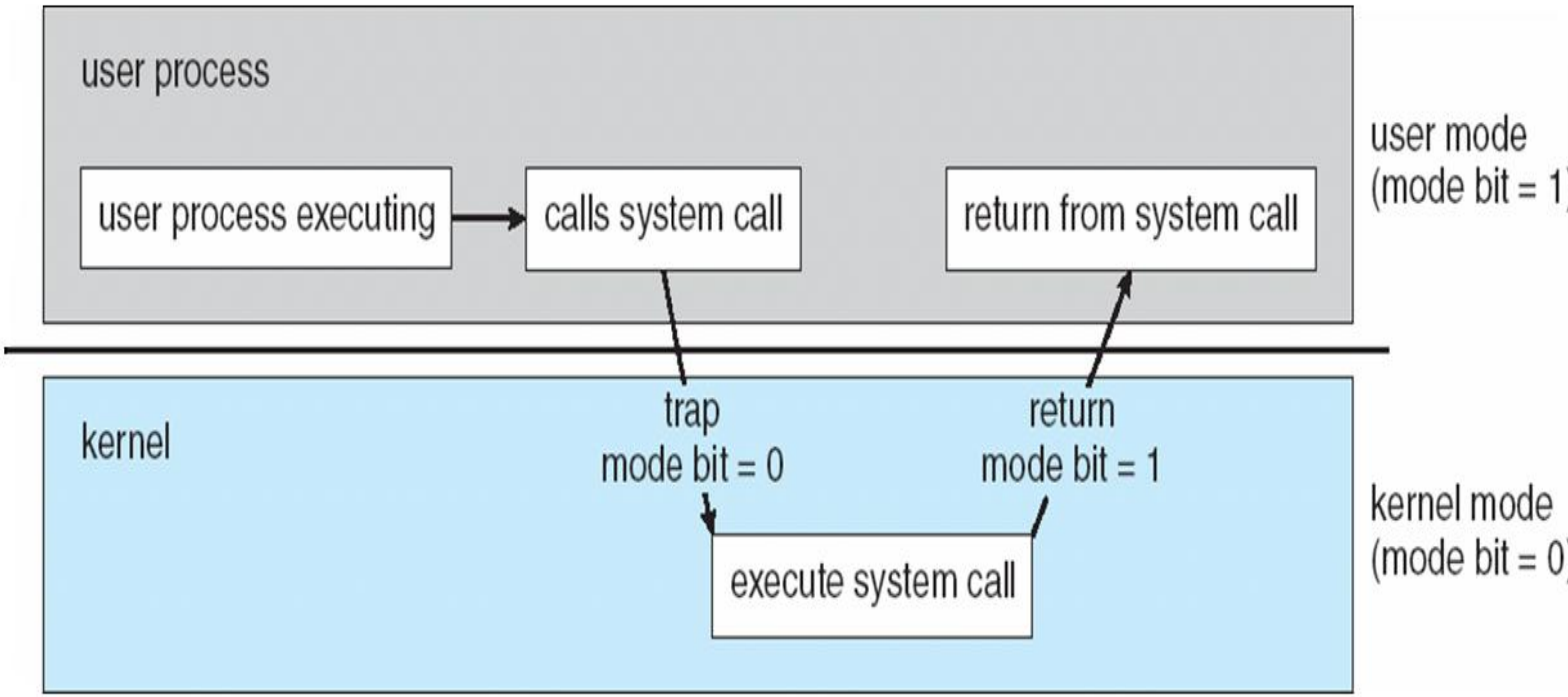


Operating-System Operations

- OS are interrupt driven.
- Since the OS and the user share the HW and SW resources , we must sure that an error in user program could cause problems only for its running.
  - EX: process problems include infinite loop, this loop could prevent the correct operation of many other processors or the OS.

- **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



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

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

Storage Management (cont.)

- **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

Storage Management (cont.)

- **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

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
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

Storage Management (cont.)

- **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

End of Chapter 1