Operating Systems Concepts (CS-351)

Week 1_A

Course Information-1

- Course: CS-351 (3 credits)
- ***Section 01:***
    - Time: Monday 7:00 PM – 9:00PM
    - Place: EC203
- ***Course Website: Canvas***

Course Information-2

- Instructor: Gouri Kar
  - Email: gkar@fullerton.edu
  - Phone: (office)
  - Office: EC203

- Office Hours:
  - By appointment

Prerequisites

- Prior to taking this course, you must either:
  - Complete all of the following courses (strictly enforced):
    - CPSC-254: UNIX and Open Source Systems
    - CPSC 301/EPP (corequisite): Programming Lab Practicum
  - Or, have the permission of the CS department.
- Failure to meet the prerequisites may result in you being dropped administratively.

Course Objectives

- To learn about the design and implementation of modern operating systems (OSs):
  - Operating Systems. What are they? What do they do?
  - Classical components of modern operating systems:
    - Processes
    - Threads
    - CPU schedulers
    - Memory managers
    - File systems
    - I/O systems
    - Inter-process communications mechanisms
  - State of the art and future:
    - Cloud computing
    - Mobile Computing

Texts

- Silberschatz, Galvin, Gagne, *Operating System Concepts, 10th edition*, ISBN: ISBN 978-1-118-06333-0.
- All additional materials shall be posted on Titanium.

Evaluation

- Course grade breakdown:
  - Programming Assignments: 30%
  - In-class or online quizzes: 15%
  - Midterm: 25%
  - Final Exam: 25%
  - Attendance and participation: 5% (may miss 1 class)
- The course grade shall be curved over the entire class, and (strictly) assigned according to the following range:
  - A+: ≥ 95
  - A: ≥ 93%
  - A-: ≥ 90%
  - B+: ≥ 88%
  - B: ≥ 82%
  - B-: ≥ 80%
  - C+: ≥ 78%
  - C: ≥ 72%
  - C-: ≥ 70%
  - D+: ≥ 68%
  - D: ≥ 62%
  - D-: ≥ 60%
  - F: <60%

Assignments

- Written assignments: To be done individually
- Programming assignments:
  - may be done by groups of 4-6 students (unless otherwise specified)
  - Must use Linux OS.
  - Must use C or C++
  - Familiarity with basic C and Unix is assumed.
  - Exceptions to these rules shall be announced.
- All completed assignments must be submitted via Canvad.
- Late assignments shall be penalized 20%.
- No assignment shall be accepted after 24 hours from the deadline.

Quizzes(2~4 totally)

- In-Class quizzes:
  - At the beginning of some classes—prepare and be ready before each class!
  - Closed book.
  - Test your understanding of the material presented in class.
  - Missed quizzes shall earn a grade of 0 (unless you can provide written evidence of a legitimate excuse e.g. doctor's note).
- Take-home quizzes:
  - Require critical thinking (and creativity!).
  - Late submissions shall be penalized 10%.
  - No quiz shall be accepted after 24 hours from the deadline.

Exams

- All exams are comprehensive and closed book.
- Midterm:
  - The midterm will take place the class around **the 8th or 9th week** (tentative).
  - There will be no make ups (unless accompanied by written evidence of a valid excuse).
- Final Exam (In-class):
  - As shown on the Admissions and Records web page (http://www.fullerton.edu/admissions/CurrentStudent/FinalExaminations.asp), the final exam will be in our lecture room:

    Saturday, May 14th 1- 2:15 PM, CS 102
  - Missed exams shall be dealt with according to University policies on incompletes and withdrawals.

Attendance and Participation

- You are required to be physically and mentally attending each class.
- The attendance is mandatory and shall be taken at the beginning of every class.
- For each attended session, the student shall earn a credit. Missed sessions shall receive 0 credit. Each student may miss one class without incurring penalties for attendance.
- Absences shall only be excused if you provide written documentation (e.g. Doctor's note) explaining the reason for the absence.
- Participate in class and online discussions (don't be afraid!).

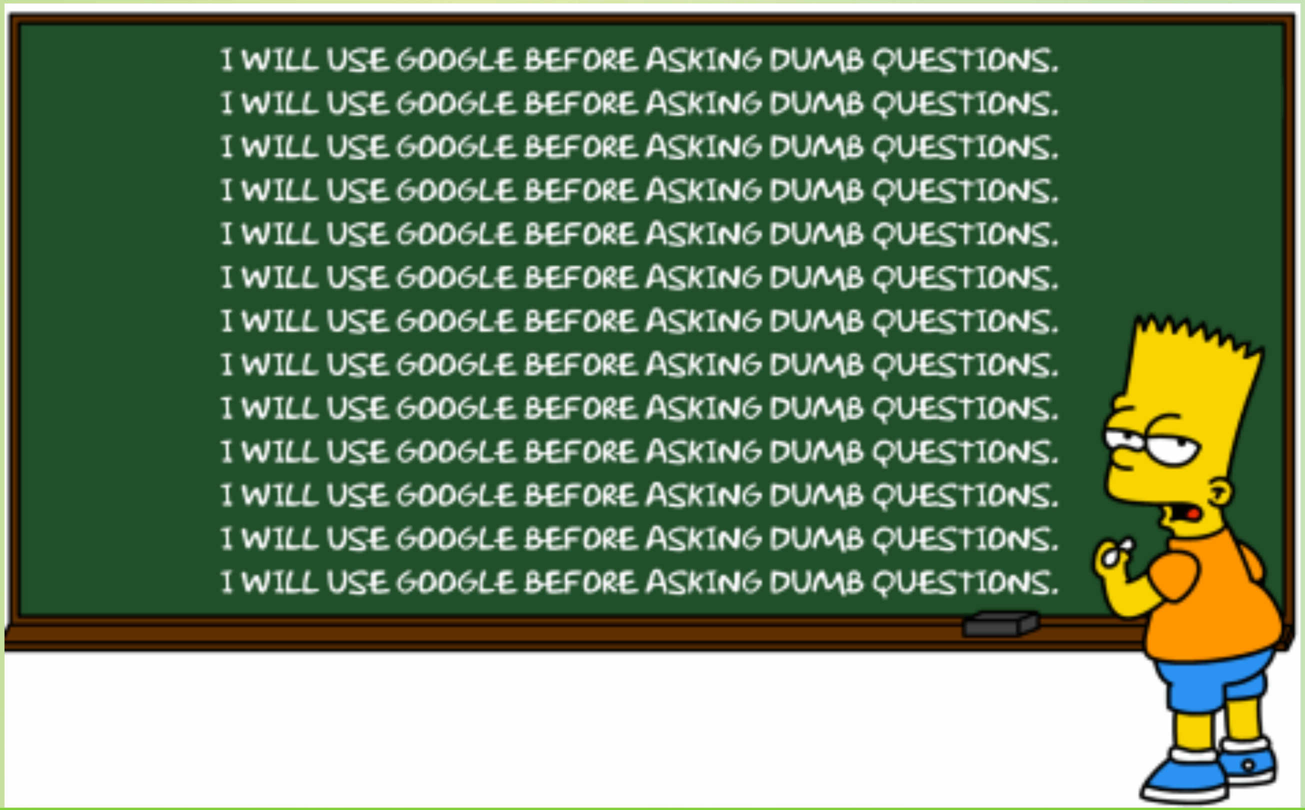
Extra Credit

- Online Discussion Forum on CANVAS-Up to Extra 2% **may be** added to your final points if you: ask or answer at least 3 questions during each class week.

- Some assignments, exams, and quizzes may include bonus sections.

Asking Questions

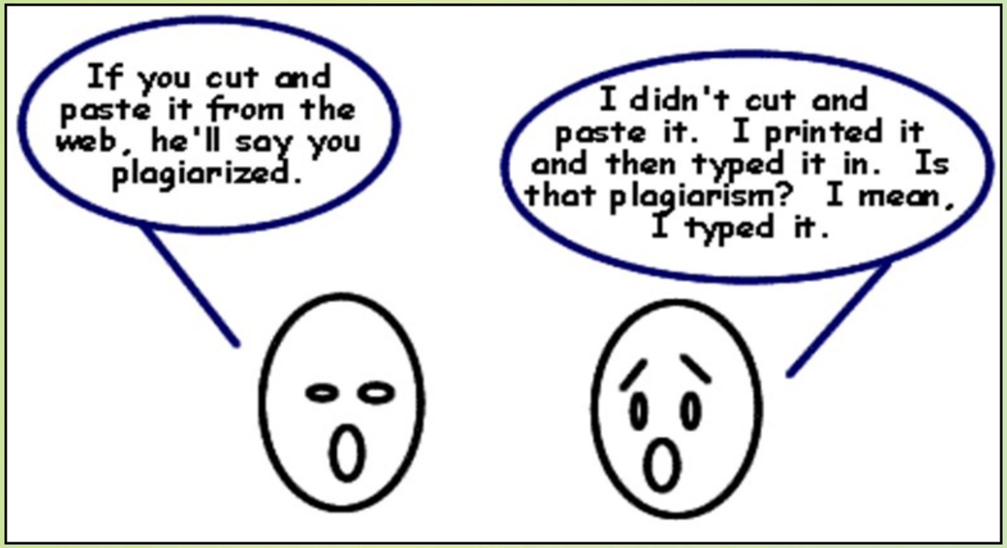
- Never be afraid to ask:
  - In class or after class.
  - During office hours.
  - Make Google your friend (can't beat the availability and response time!)


I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.
I WILL USE GOOGLE BEFORE ASKING DUMB QUESTIONS.

Class Cancellation Policy

- All class cancellations shall be announced by email.
- Instructor does not arrive within the first 15 minutes of class = class is canceled.

Academic Honesty


If you cut and paste it from the web, he'll say you plagiarized.
I didn't cut and paste it. I printed it and then typed it in. Is that plagiarism? I mean, I typed it.

THIS IS JUST NOT GOOD ENOUGH!
SHE'S PLAGIARISED!
BUT... IT'S FROM YOUR BOOK!

Plagiarism is a totally new concept for many students. In some cultures using the words of the experts is preferable.

Academic Honesty

- Incidents of cheating shall be treated with utmost seriousness.
- You may discuss the problems with other students, however, you must write your own solutions.
- Discussing solutions to the problem is NOT acceptable.
- Copying an assignment from another student or allowing another student to copy your work may lead to an automatic F for this course.
- If you have any questions about whether an act of collaboration may be treated as academic dishonesty, please consult the instructor before you collaborate.
- Moss shall be used to detect plagiarism in programming assignments.

Emergency Policy

- Please familiarize yourself with the actions to take in case of an emergency.
- The information can be found at http://prepare.fullerton.edu/

Disabled Student Services

- Information for students with disabilities can be found at: http://www.fullerton.edu/DSS/

Course Syllabus

- You are required to read the syllabus!
- A copy of the syllabus is available on Titanium.
- If something is not clear, ask the instructor.

Introduction to Operating Systems

Chapter 1

Agenda

- Operating Systems, what are they? What do they do?
- Computer organization fundamentals.
- Operating System Functions
- Process Management
- Memory Management
- Storage Management
- I/O Management
- Protection and Security
- Distributed Systems
- Special Purpose Operating Systems
- Computing Models

What is an Operating System?

- A program that:
  - Runs at all times (a.k.a. a resident monitor, a.k.a. kernel)
  - Manages computer's hardware resources.
  - Provides basis for application programs.
  - Acts as an intermediary between user and hardware.
  - No completely adequate definition.

- Why do we need operating systems?
  - Increase the usability of computers.
  - Simplify problem solving.

Enter the World of Operating Systems

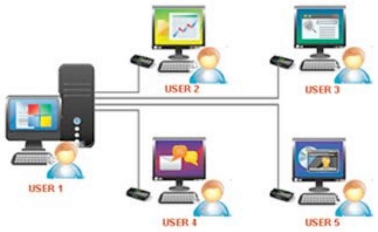
Perspectives and Roles of OS

User's perspective

OS role is to make computers easier to use and simplify problem solving

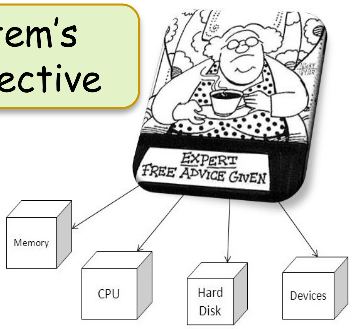
Most users want convenience, **ease of use** and **good performance**

- Interact with OS using keyboard, monitor, mouse and GUI
- Don't care about **resource utilization**



Multiple users connect to a mainframe computer using terminals
- Want systems that are designed to maximize **resource utilization** and **fair sharing** of resources among users

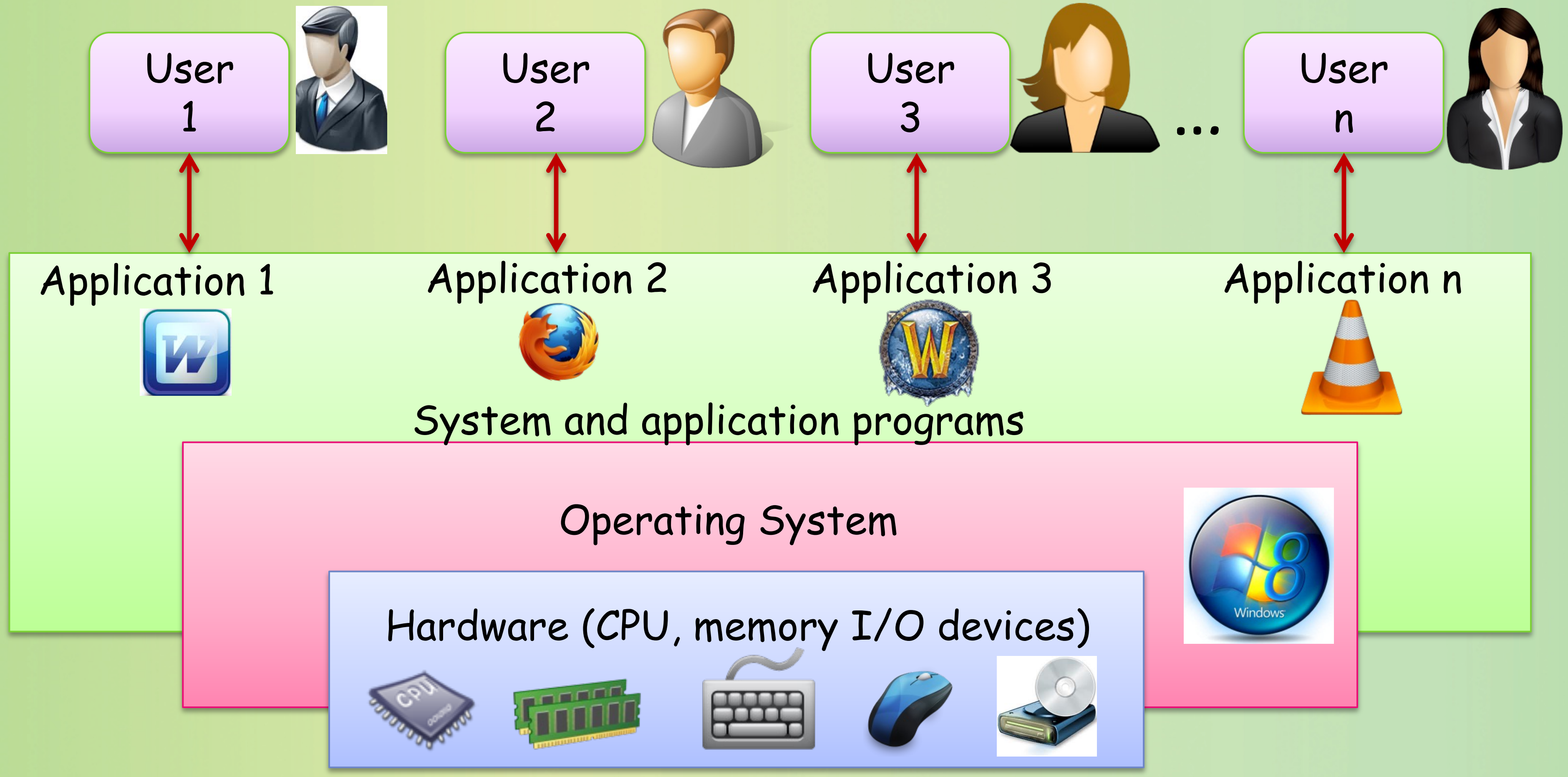
System's perspective



- Resource allocator: allocates and manages hardware resources: CPU time, memory space, disk space, and I/O
- Control program: manages the execution of user programs

What does an operating system do?

- Components of a computer system: application programs, operating system, and computer hardware.



What does an operating system do?

- Overall: provides the means for proper use of system resources e.g. hardware, software, and data.
  - Similar to a government, OS serves no useful function when by itself. It only provides an environment in which user programs can do useful work.

What does an operating system do?

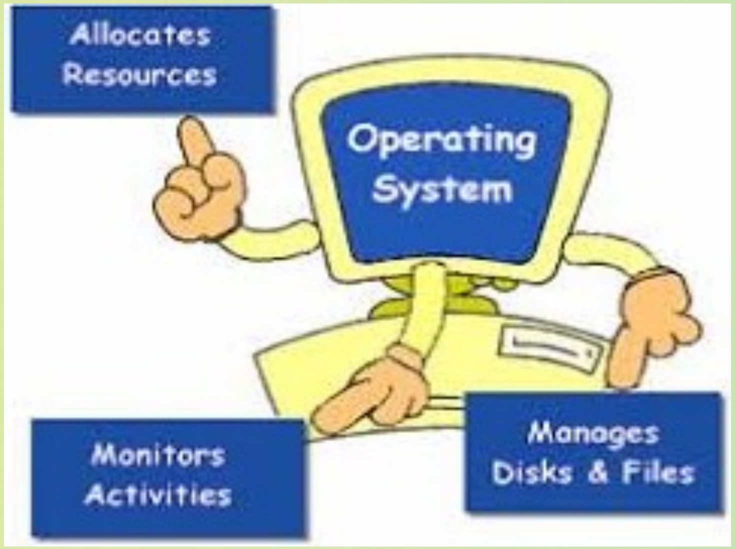
- User perspective of OS:
  - Generally:
    - Makes computers easier to use.
    - Simplifies problem solving.
  - Varies according to the interface being used.
    - Example: desktop OSs focus on ease of use and performance; enhance productivity (or play).
    - Example: work station OSs balance usability against resource utilization (i.e. how hardware and software are shared).

What does an operating system do?

- System Perspective:
  - Resource allocator: allocates and manages hardware resources e.g. CPU time, memory space, disk space, and I/O.
  - Control program: manages the execution of user programs.

Operating System Definition Summary

- No universally accepted definition!
- A program running at all times on the computer ➔ the kernel
    - Manages computer's hardware resources
    - Provides basis for application programs
    - Acts as an intermediary between user and hardware



Overview of OS Functions

- Process Management
- Memory Management
- Storage Management
- I/O Management
- Protection and Security

Process Management

- Process: a unit of work on the system
- Program vs. Process:
  - Program: is a set of instructions (a passive entity).
  - Process: a program in execution (an active entity).
- Processes require:
  - CPU time
  - Memory
  - Files
  - I/O devices

Process Management

- Operating system must:
  - Allocate resources when the process starts, manage them while it runs, and reclaim them when the process terminates.
  - Multiplex resources among multiple processes.
  - Provide means for suspending/resuming a process.
  - Provide means for process synchronization.
  - Provide means for process communications.
  - Manage Deadlock handlings.

Process Management – OS as *intermediary*

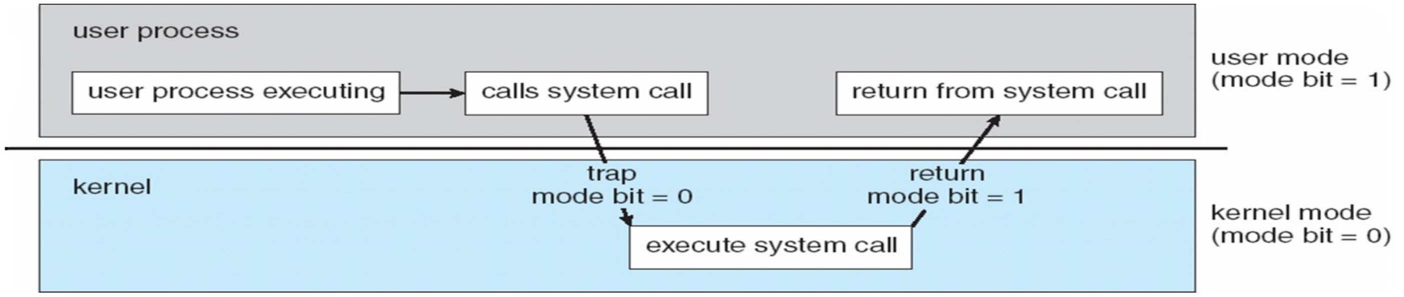
- Acts as an intermediary between the process and the rest of the system
  - Restricts processes from accessing hardware directly
  - If a process needs to access hardware such as opening a file on the disk, the process must request OS to perform the access on the process's behalf
    - Shifts the burden of dealing with hardware characteristics from the application developers to the OS
    - Allow OS to enforce order, e.g., deny invalid/unauthorized accesses

Process Management - System Calls

- OS exposes a set of system calls - functions which processes can use to request services from the operating system
  - Read file from the disk
  - Send data over the network
  - Send message to another process
- Example
  - sys_open(): opens a file
  - sys_close(): closes the file
  - sys_read(): reads from file
  - sys_write(): writes to file
- Linux system calls are defined in part of the operating system, the system call table

Process Management – Dual Mode Operation

- A typical computer supports two modes of execution:
  - User mode (unprivileged mode) - when a process is executing
    - Allows the process to execute only unprivileged instructions e.g., addition, subtraction, logical operations, …
    - CPU ***restricts*** execution of privileged instructions (e.g., instructions for directly accessing hardware, managing OS timers, …)
  - Kernel/system mode (privileged mode) - when the OS is executing
    - CPU ***allows*** execution of privileged instructions



Overview of OS Functions

- Process Management
- Memory Management
- Storage Management
- I/O Management
- Protection and Security

Memory Management

- Main memory:
  - A large array of bytes or words where each word or byte has its own address.
  - The only large storage directly accessible by the CPU.
- For improved resource utilization, several programs are usually kept in memory:
  - Introduces the need for memory management.
- Memory Management:
  - Keep track of what parts of memory are being used by what processes.
  - Decide which processes (or parts of) to move into and out of memory.
  - Allocate and deallocate memory as needed.

Overview of OS Functions

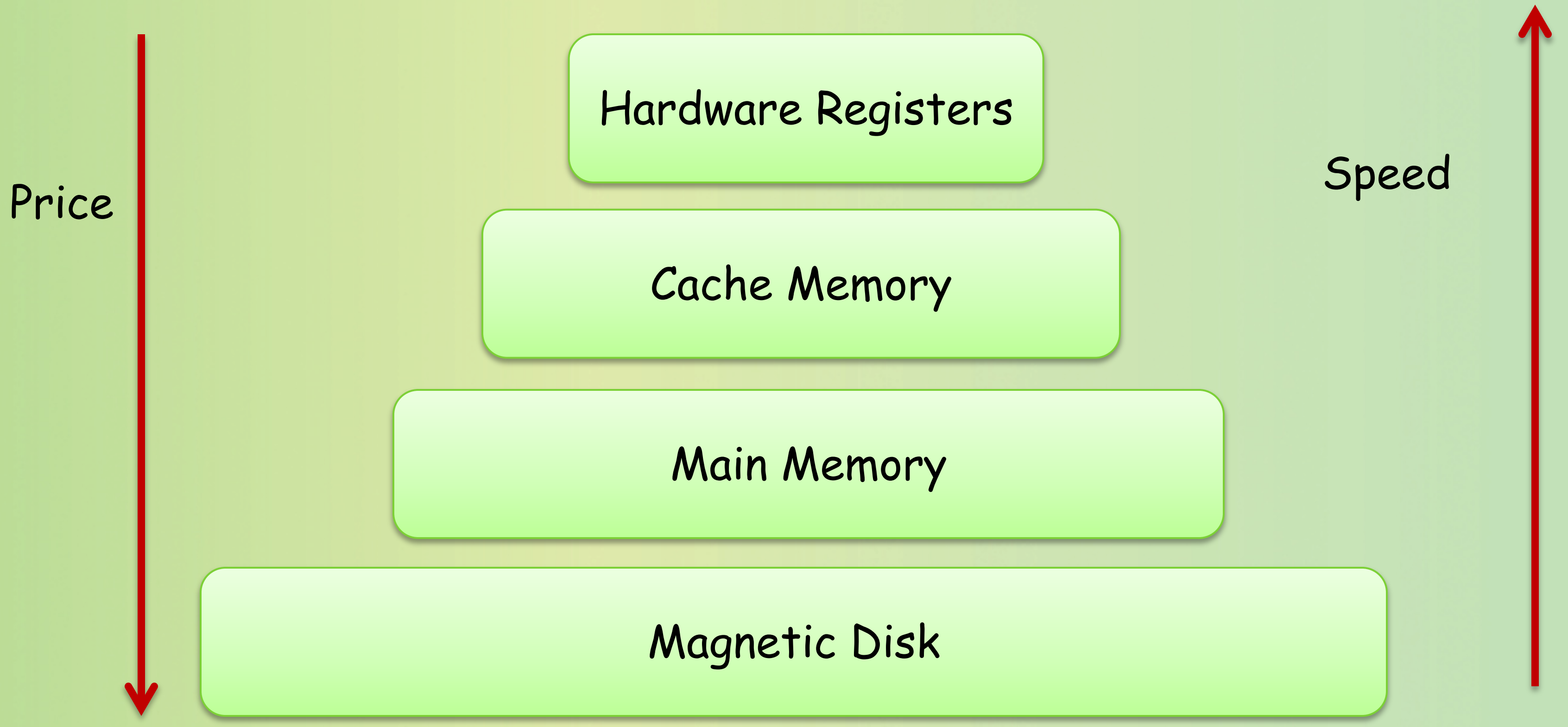
- Process Management
- Memory Management
- Storage Management
- I/O Management
- Protection and Security

Storage Management

- For convenience, the information on the storage device is abstracted into units called files.
- File is a collection of related information defined by its creator.
- Operating system file management services:
  - Creating and deleting files.
  - Creating and deleting directories (i.e. collections of files).
  - Mapping files to memory on the storage device.
  - File Backup.
- Mass-storage (e.g. disk) management:
  - Managing free space
  - Storage allocation
  - Disk scheduling (i.e. managing multiple operations that read/write to/from the disk).

Storage Management

- Caching: temporarily store data/instructions in faster storage (i.e. cache) and access them from there.



- Must ensure data consistency along all levels of the hierarchy.

Overview of OS Functions

- Process Management
- Memory Management
- Storage Management
- I/O Management
- Protection and Security

I/O Management

- Purpose: hide the peculiarities of specific devices from the user.
- Example: the Unix I/O subsystem provides:
  - Functionality to manage data buffering, caching, and spooling.
  - A general device driver interface.
  - Drivers for specific hardware devices (which know how to control the specific device).

Overview of OS Functions

- Process Management
- Memory Management
- Storage Management
- I/O Management
- Protection and Security

Protection and Security

- Protection: controls the access of users and processes to the system resources (e.g. memory, files, etc).
  - Internal to the OS
  - Example: process A attempts to (illegally) write to the memory of process B. The operating system detects the violation and terminates process A.
- Security: defending the OS against external threats.
  - Example: malware (e.g. viruses, worms, etc).
- Protection vs. Security: definitions vary

Types of Operating Systems

- ❑ Distributed Systems
- ❑ Mobile Computing
- ❑ Real-time Embedded Systems
- ❑ Multimedia Systems
- ❑ Virtual Machines
- ❑ Cloud Computing

Computing Environments: Distributed Systems

- A collection of physically separate, networked systems that share resources.
- Advantages of resource sharing: increases computation speed, functionality, data availability, and reliability.
- Types of networks:
  - Local-Area Network (LAN): connects systems within a single floor room or building.
  - Wide-Area Network (WAN): connects buildings, cities, or countries.

Computing Environments: Mobile Computing

- Computing on handheld smartphones and tablet computers.
- Sacrifice screen size, memory capacity, CPU power, in favor of portability:
  - This will change as mobile devices grow more powerful.
- Allows for applications impractical on traditional systems (e.g. laptops, desktops, etc).
  - Navigation
  - Augmented reality
  - Many others!
- Dominant OSs:
  - iOS: designed for Apple's iPhone and iPad platforms.
  - Android: runs on all sorts of devices.

Real-Time Embedded Systems

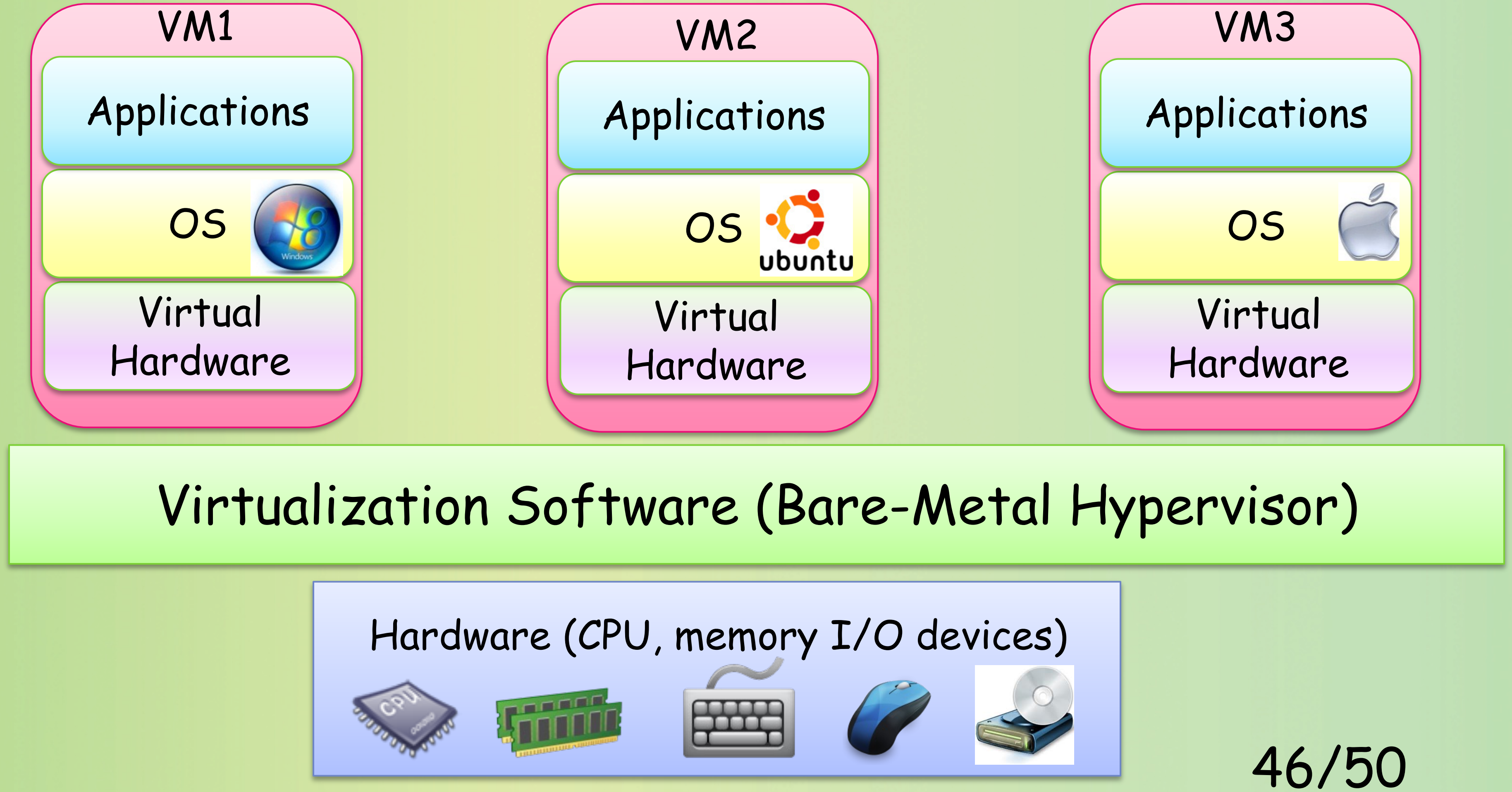
- Embedded Systems: systems dedicated to specific tasks e.g. controlling car engines, robotic arms, etc.
  - Usually have little to no user interface.
  - One of the most prevalent types of computers.
- Real-Time Systems: systems where the processing must be done within the defined timing constraints.
  - Example: the robotic arm must stop moving before it smashes into a car it was building (not after).
  - Embedded systems usually run real-time operating systems.

Multimedia Systems

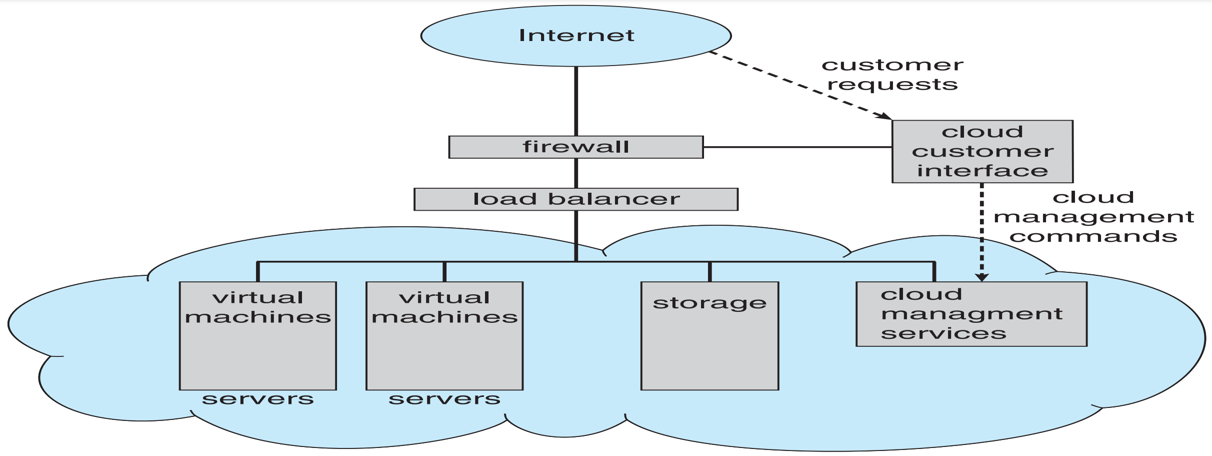
- Systems specialized to deliver media content (i.e. a mix of text, video, sound, etc).
- Some media must be delivered within certain timing constraints.

Computing Models: Virtual Machines

- Virtual Machines (VMs): a software that can run its own operating system and applications just like a real physical system.



Cloud Computing (1)



Delivers services over the network

- Three types of cloud computing services:
  - Infrastructure-as-a-Service (IaaS): provides hardware resources: storage, CPU, and networking services
  - Platform-as-a-Service (PaaS): provides hardware and platform ready for applications (e.g., a database server)
  - Software-as-a-Service (SaaS): provides software applications over the network
- Helps reduce operating costs
- Helps to improve resource utilization (by combining computing resources)
- Makes it easier to tackle large scale computing problems

Computing Models: Cloud Computing (2)

- Types of clouds:
  - Public: available to anybody willing to pay for the services.
    - Example: Amazon's Elastic Cloud (EC2)
    - Example: Google's App Engine
  - Private: a cloud run by a company for its own use.
    - Example: IBM SmartCloud Foundation
  - Hybrid: a combination of public and private.

Next Class

- Chapter 2: Processes

Tasks before next class:

- C/C++ programming preparation
- Linux/Unix programming
- Complete the reading of Chapters 1, 2 & 3