CSCI315 – Operating Systems Design

Department of Computer Science
Bucknell University

Introduction

Ch 1

This set of notes is based on notes from the textbook authors, as well as L. Felipe Perrone and other instructors.
Xiannong Meng, Fall 2021.

What is an Operating System?

What happens when we ask a computing device such as laptop or smartphone to do something for us?

Photo by UX Indonesia on Unsplash

- You type a command such as “gcc -o hello hello.c” or you click a button such as “send.”

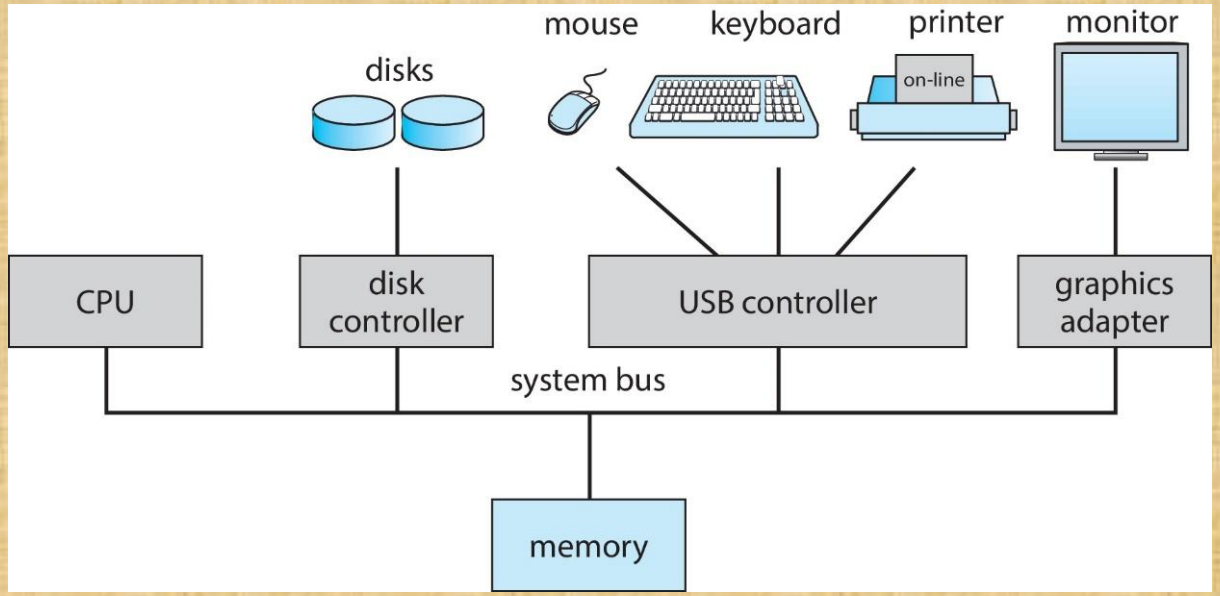
- The following will be done by the computer (roughly). Take the “gcc” command above as an example.
  - Read and interpret your command.
  - Read your hello.c file from the disk into memory.
  - Compile your program into object code.
  - Link your object code with appropriate libraries.
  - Generate and save the executable, which is a file, back to the disk.

- Now imagine 30 or so other students are doing exactly the same thing as you are in a lab setting.
  - How a single computer such as **linuxremote.bucknell.edu** handle this?

What does an operating system do

- Operating systems such as Linux, Windows, MacOS, Android work with a collection of **systems and applications software** such as gcc, browsers, text editors, Word, and many, many others to provide the users with an effective and useful computing environment to accomplish user tasks.
- Operating systems are themselves a piece of software that interacts among the hardware, the applications software, and the users.
- They are a **resource manager** that allocate resources such as memory, disk, and CPU time to applications.

A Modern Computer System

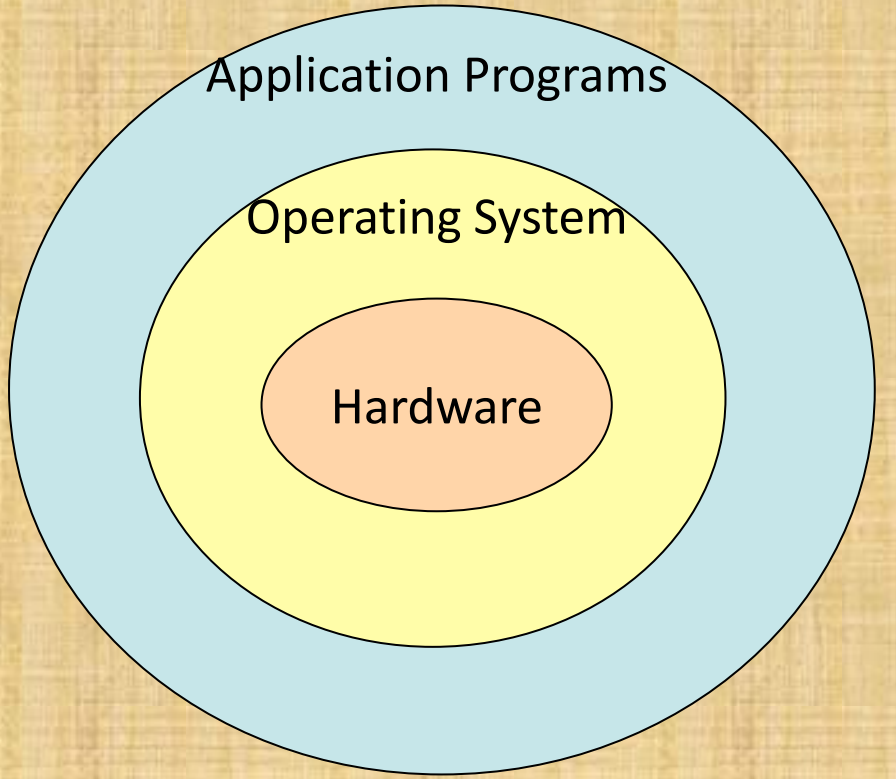


The blocks shown are hardware components. Software runs on these components.

Computer System Components

1. **Hardware** – provides basic computing resources (CPU, memory, I/O or Input and Output devices).
2. **Operating system** – controls and coordinates the use of the hardware among the various application programs for different users.
3. **Applications programs** – define the ways in which the system resources are used to complete the computing tasks of the users (compilers, database systems, video games, business programs).
4. **Users** (people, machines, other computers).

Macroscopic Abstract View of the Computer System

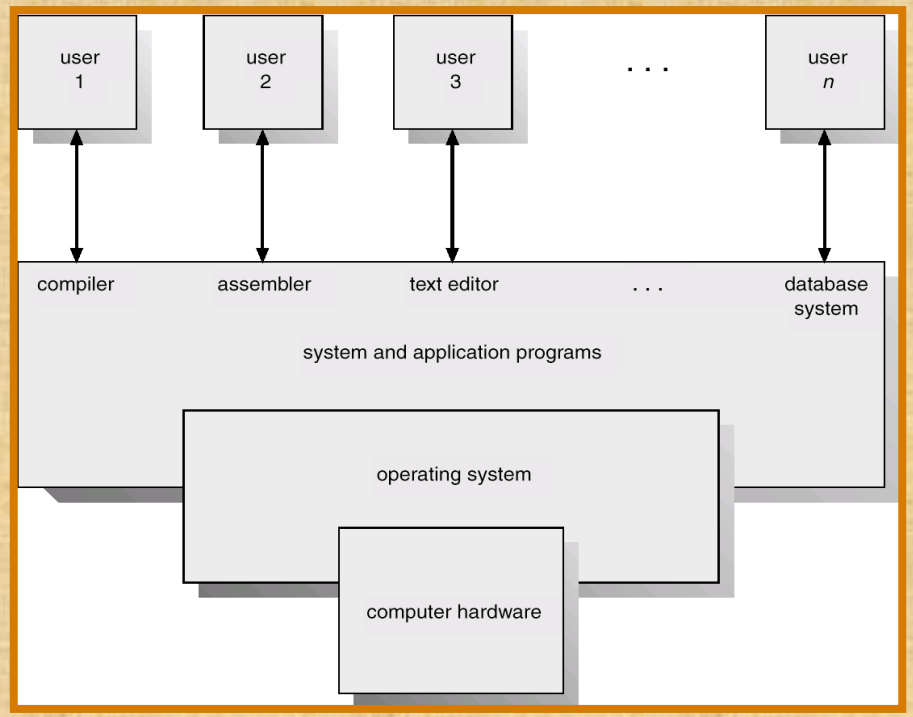


What does an Operating System do?

A "program" that acts as an intermediary between a user of a computer and the computer hardware.

- Manage resources in the computer system
- Control the execution of programs

Abstract View of System Components



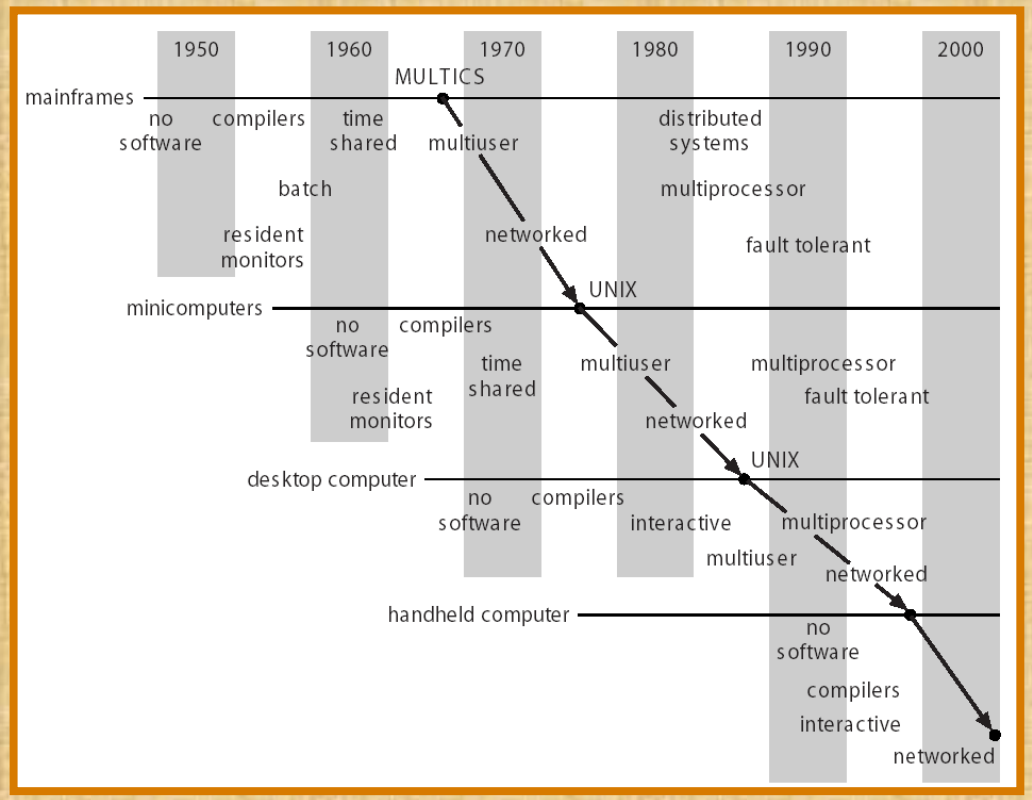
Operating System Definitions

An operating system can be defined in any of the three ways.

- **Resource allocator** – manages and allocates resources.
- **Control program** – controls the execution of user programs and operations of I/O devices.
- **Kernel** – the one program “running” at all times (all else being application programs).

OPERATING SYSTEMS EVOLUTION

OS Evolution Timeline



Cloud computing in 2010s and 2020s

Mainframes

- Centralized computing;
- Users submit jobs through “batched” system;
- Multiple jobs take turns to run – the operating system decides which one to run at what time;
- No direct user interaction.

Time-Sharing Systems and Interactive Computing

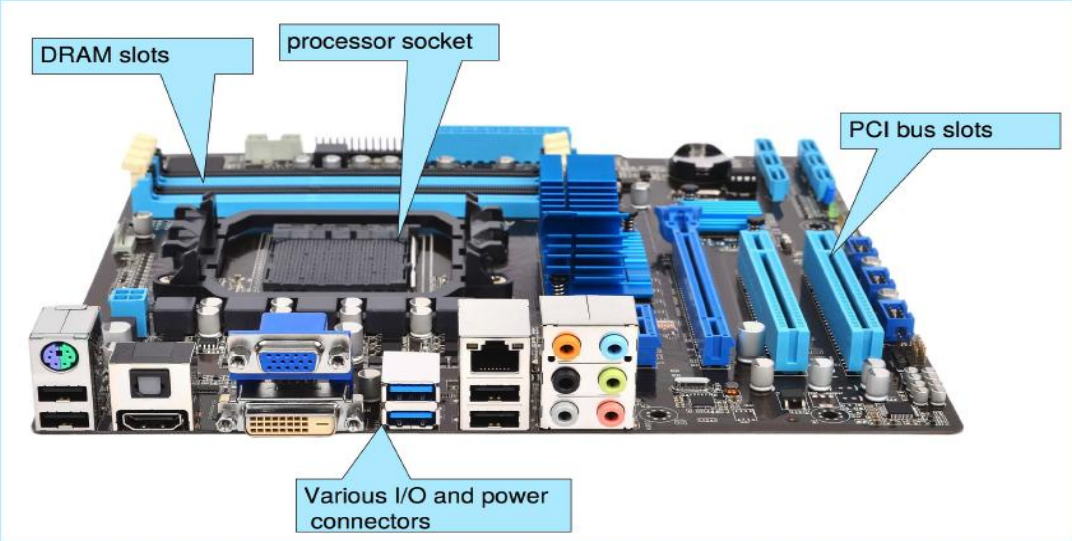
- The CPU is multiplexed among all (usually many) jobs that "time share" the computer.
- Computing jobs may be swapped in and out between memory and disk.
- Interactive communication between the user and the system is provided:
  - When the operating system finishes the execution of one command, it seeks the next "control statement" from the user's keyboard
- Interactive communication must be available for users to access data and code. (Another way to communicate with the system is through "batch" jobs.)

Desktop Systems

- **Personal computers** – computer system dedicated to a single user.
  - I/O devices – keyboards, mice, display screens, small printers.
  - User convenience and responsiveness.
  - Technology adopted from larger operating system:
    - Often individuals have the sole use of the computer and do not need advanced CPU utilization of protection features.
  - May run several different types of operating systems (Windows, MacOS, UNIX, Linux).

PC Motherboard

Consider the desktop PC motherboard with a processor socket shown below:

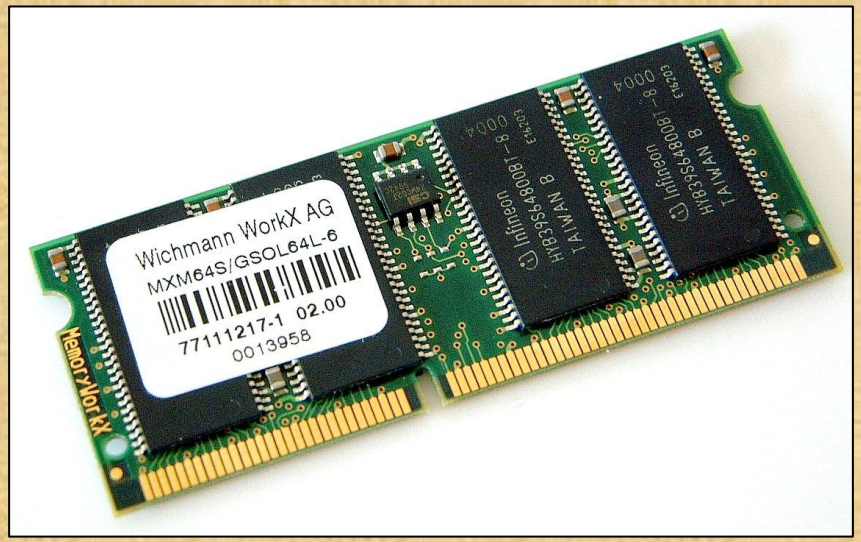


This board is a fully-functioning computer, once its slots are populated. It consists of a processor socket containing a CPU, DRAM sockets, PCIe bus slots, and I/O connectors of various types. Even the lowest-cost general-purpose CPU contains multiple cores. Some motherboards contain multiple processor sockets. More advanced computers allow more than one system board, creating NUMA systems.

Intel Core i9 Processor (CPU)

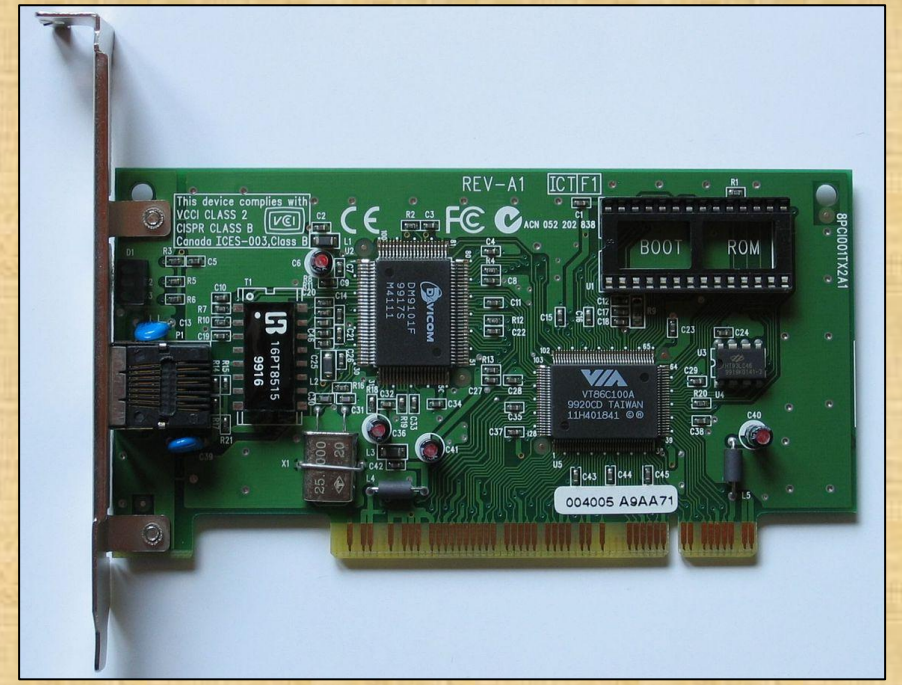
https://en.wikipedia.org/wiki/File:Intel_Core_i9-10900K.png

SDRAM Board Example



https://commons.wikimedia.org/wiki/File:SODIMM_64MB_SDRAM.JPG

PCI Board Example: Fast Ethernet

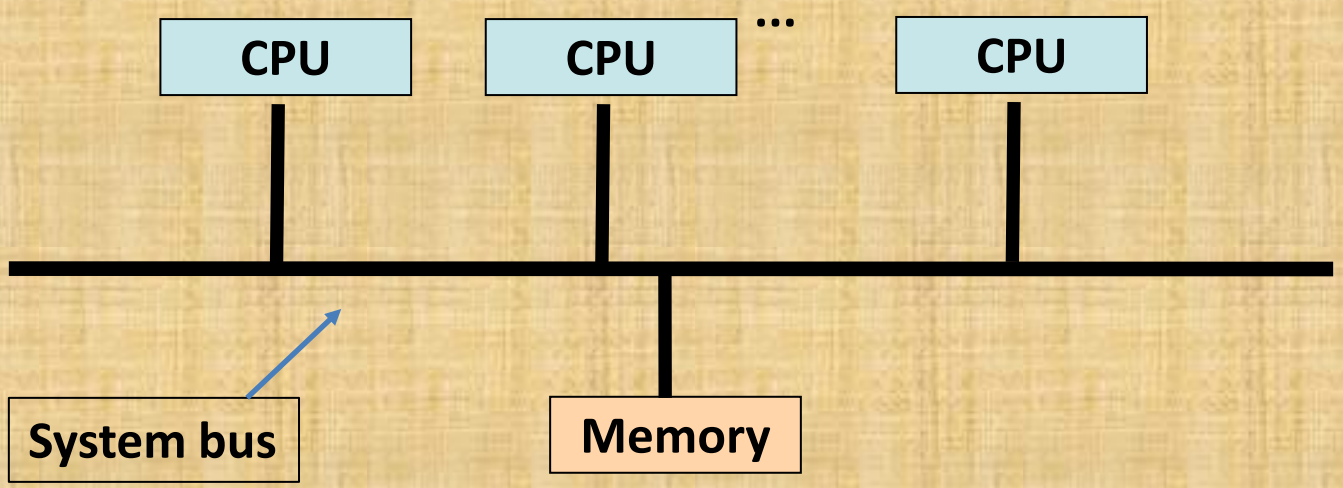


https://commons.wikimedia.org/wiki/File:Fast_Ethernet_PCI_Network_Interface_Card_SN5100TX_(VIA_VT86C100A).jpg

Parallel Systems

- Systems with more than one CPU in close communication (a.k.a. *multiprocessor systems*), or more than one core on a single processor, a.k.a. *multi-core* systems.
- **Tightly coupled system** – processors share memory and a clock; communicate through the shared memory.
- Advantages of parallel system:
  - Increased *throughput*
  - Economical
  - Increased reliability (in some cases)
    - graceful degradation
    - fail-soft systems

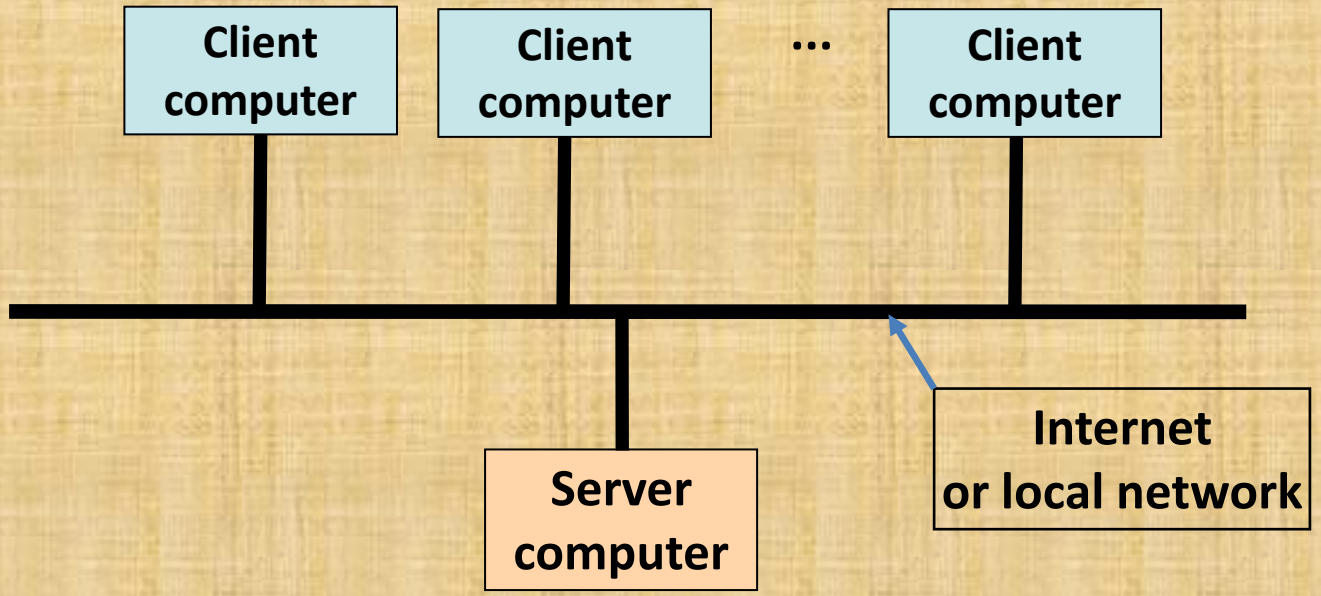
Symmetric Multiprocessor Architecture



Distributed or Networked Systems

- Distribute the computation among several physical processors.
- **Loosely coupled system** – each processor has its own local memory; processors communicate with one another through various communications medium, such as high-speed buses or internet.
- Advantages of distributed systems:
  - Resources Sharing;
  - Computation speed up – load sharing;
  - Reliability;
  - Communications.
- In the lab work we do, you will experiment with this model (client and server)

Networked Systems Architecture (client-server model)



Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - –Usually sharing disk 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**
  - –Our Linux systems such as **linuxremote** can be considered as an example of clustered system.

Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- For example, a computing system that controls the launch and flight of a space-craft.

Embedded Systems

- Appliances, smart sensors, digital control systems.
- Issues:
  - Limited memory;
  - Slower processors;
  - Small display screens (if any).
- For example, our classroom temperature control system, or the computers in your car or microwave oven.