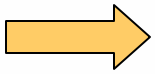
Introduction

- We first need to explore the fundamentals of computer processing
  - components of a computer
  - how those components interact
  - how computers store and manipulate information
  - computer networks
  - the Internet and the World Wide Web
  - programming and programming languages

Outline



Computer Processing

Hardware Components

Networks

Programming Languages

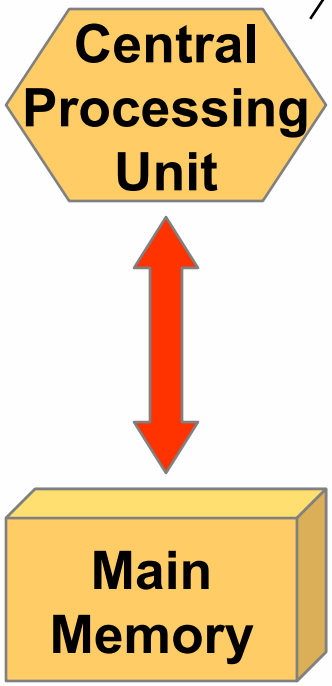
Hardware and Software

- **Hardware**
  - the physical, tangible parts of a computer
  - keyboard, monitor, disks, wires, chips, etc.
- **Software**
  - programs and data
  - a *program* is a series of instructions
- **A computer requires both hardware and software**
- **Each is essentially useless without the other**

The key hardware components

- Central processing unit
  - Executes the individual commands of a program
- Input/output (I/O) devices
  - Allow a human being to interact with the computer
- Main memory
  - Holds the software while it is being processed by the CPU
- Secondary memory devices
  - Store software in a relatively permanent manner

CPU and Main Memory



Chip that executes program commands

Intel Pentium 4
Sun ultraSPARC III

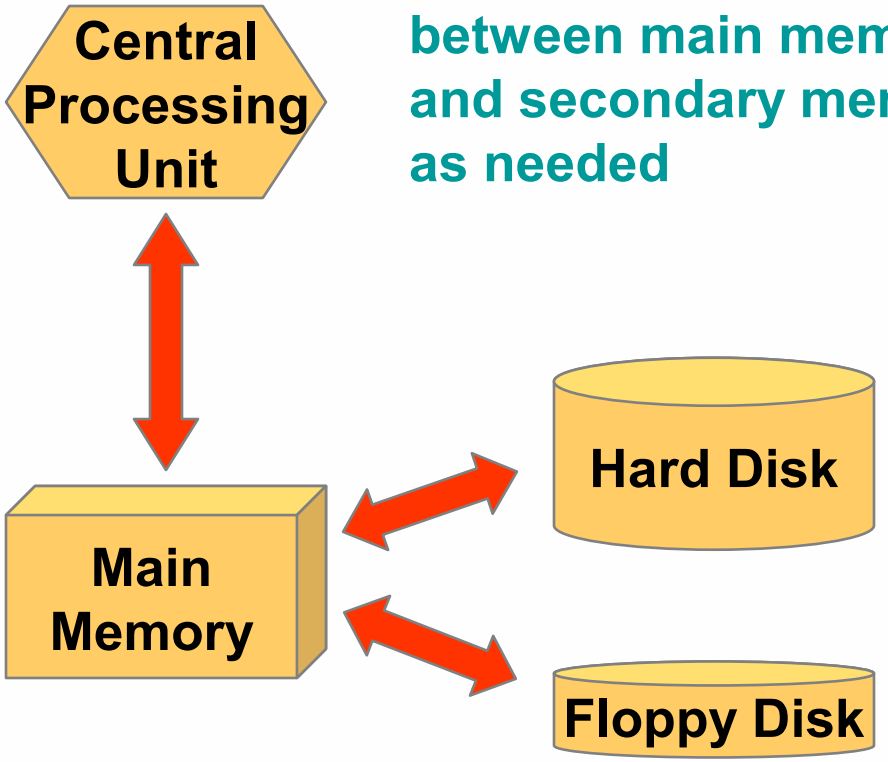
Primary storage area for programs and data that are in active use

Synonymous with RAM

Secondary Memory Devices

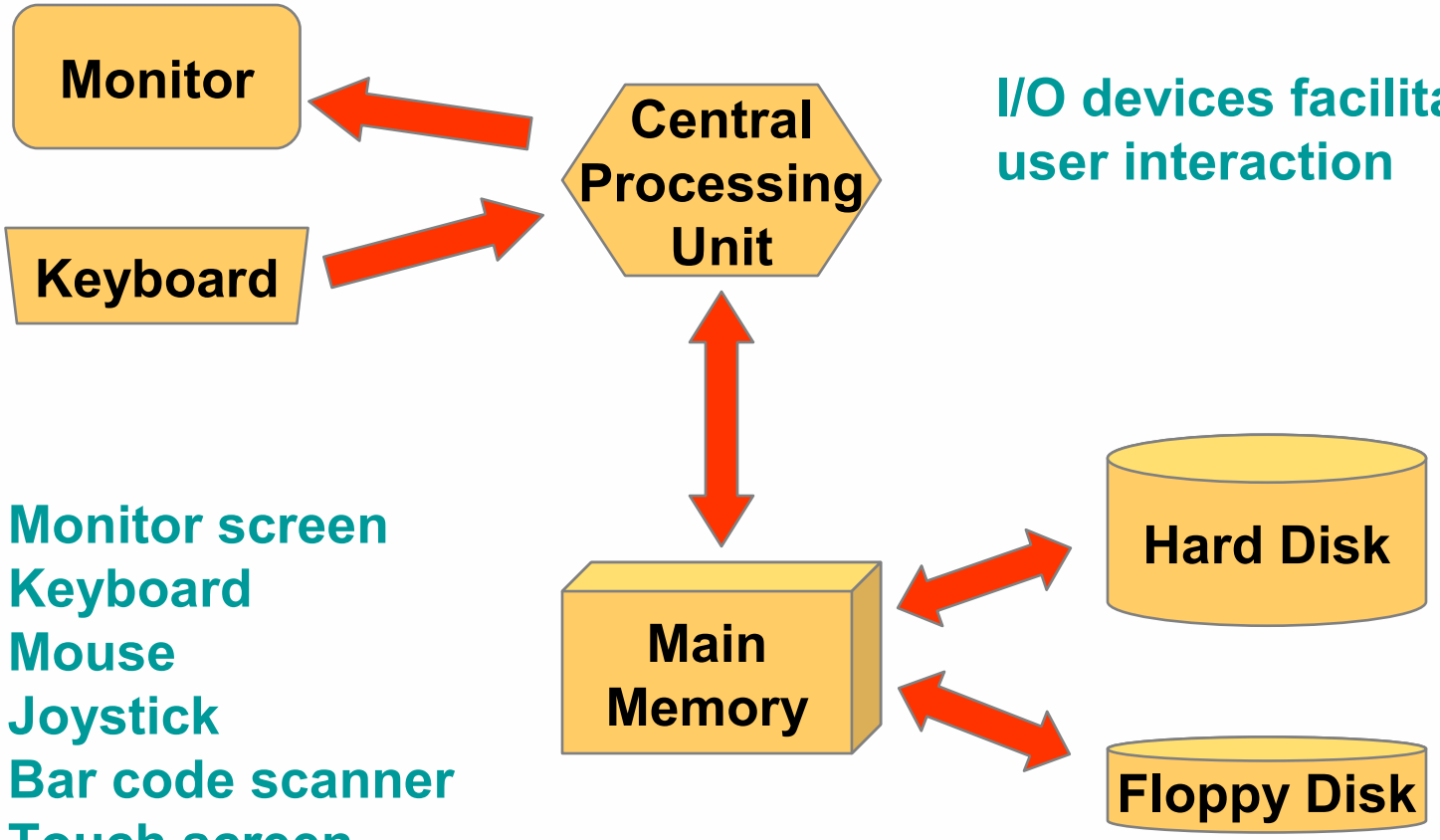
Secondary memory devices provide long-term storage

Information is moved between main memory and secondary memory as needed



Hard disks
Floppy disks
ZIP disks
Writable CDs
Writable DVDs
Tapes

Input / Output Devices



I/O devices facilitate user interaction

Monitor screen
Keyboard
Mouse
Joystick
Bar code scanner
Touch screen

Software Categories

- **Operating System**
  - controls all machine activities
  - provides the user interface to the computer
  - manages resources such as the CPU and memory
  - Windows XP, Unix, Linux, Mac OS
- **Application program**
  - generic term for any other kind of software
  - word processors, missile control systems, games
- **Most operating systems and application programs have a *graphical user interface* (GUI)**

Analog vs. Digital

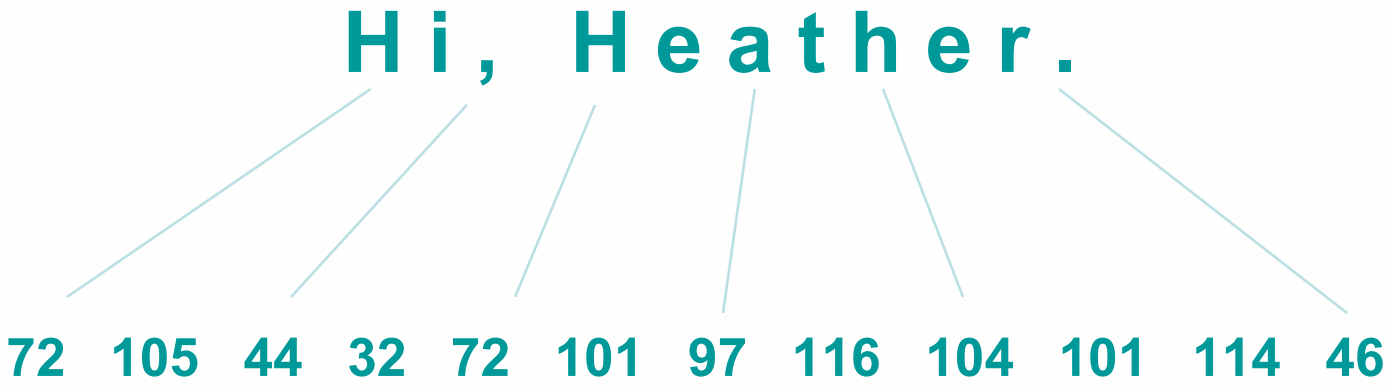
- There are two basic ways to store and manage data:
- *Analog*
  - continuous, in direct proportion to the data represented
  - music on a record album - a needle rides on ridges in the grooves that are directly proportional to the voltages sent to the speaker
- *Digital*
  - the information is broken down into discrete pieces, and each piece is represented separately
  - music on a compact disc - the disc stores numbers representing specific voltage levels sampled at specific times

Digital Information

- **Computers store all information digitally:**
  - **numbers**
  - **text**
  - **graphics and images**
  - **video**
  - **audio**
  - **program instructions**
- **In some way, all information is *digitized* - broken down into pieces and represented as numbers**

Representing Text Digitally

- For example, every character is stored as a number, including spaces, digits, and punctuation
- Corresponding upper and lower case letters are separate characters



Binary Numbers

- Once information is digitized, it is represented and stored in memory using the *binary number system*
- A single binary digit (0 or 1) is called a *bit*
- Devices that store and move information are cheaper and more reliable if they have to represent only two states
- A single bit can represent two possible states, like a light bulb that is either on (1) or off (0)
- Permutations of bits are used to store values

Bit Permutations

1 bit	2 bits	3 bits	4 bits	
0	00	000	0000	1000
1	01	001	0001	1001
	10	010	0010	1010
	11	011	0011	1011
		100	0100	1100
		101	0101	1101
		110	0110	1110
		111	0111	1111

Each additional bit doubles the number of possible permutations

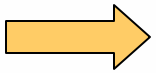
Bit Permutations

- Each permutation can represent a particular item
- There are 2^N permutations of N bits
- Therefore, N bits are needed to represent 2^N unique items

How many items can be represented by

1 bit ?	$2^1 = 2$ items
2 bits ?	$2^2 = 4$ items
3 bits ?	$2^3 = 8$ items
4 bits ?	$2^4 = 16$ items
5 bits ?	$2^5 = 32$ items

Outline



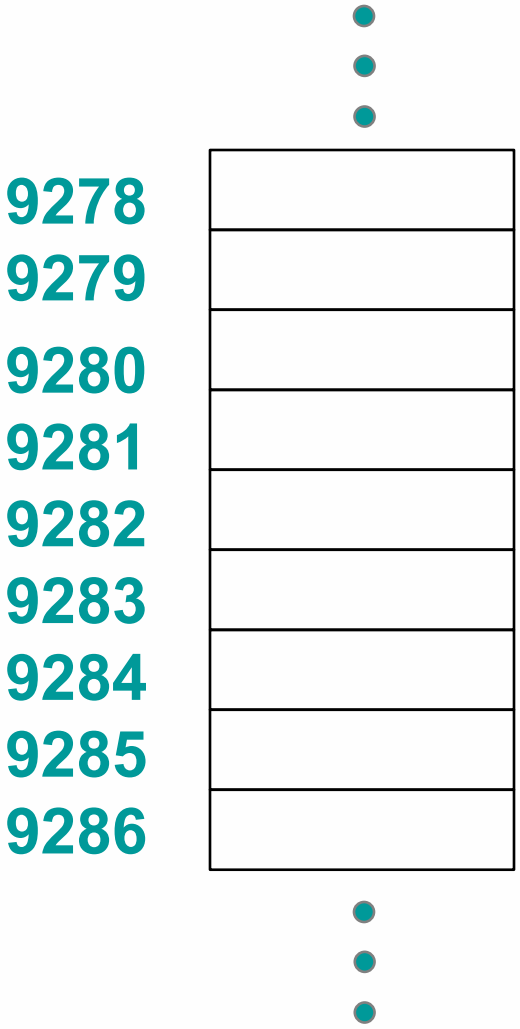
Computer Processing

Hardware Components

Networks

Programming Languages

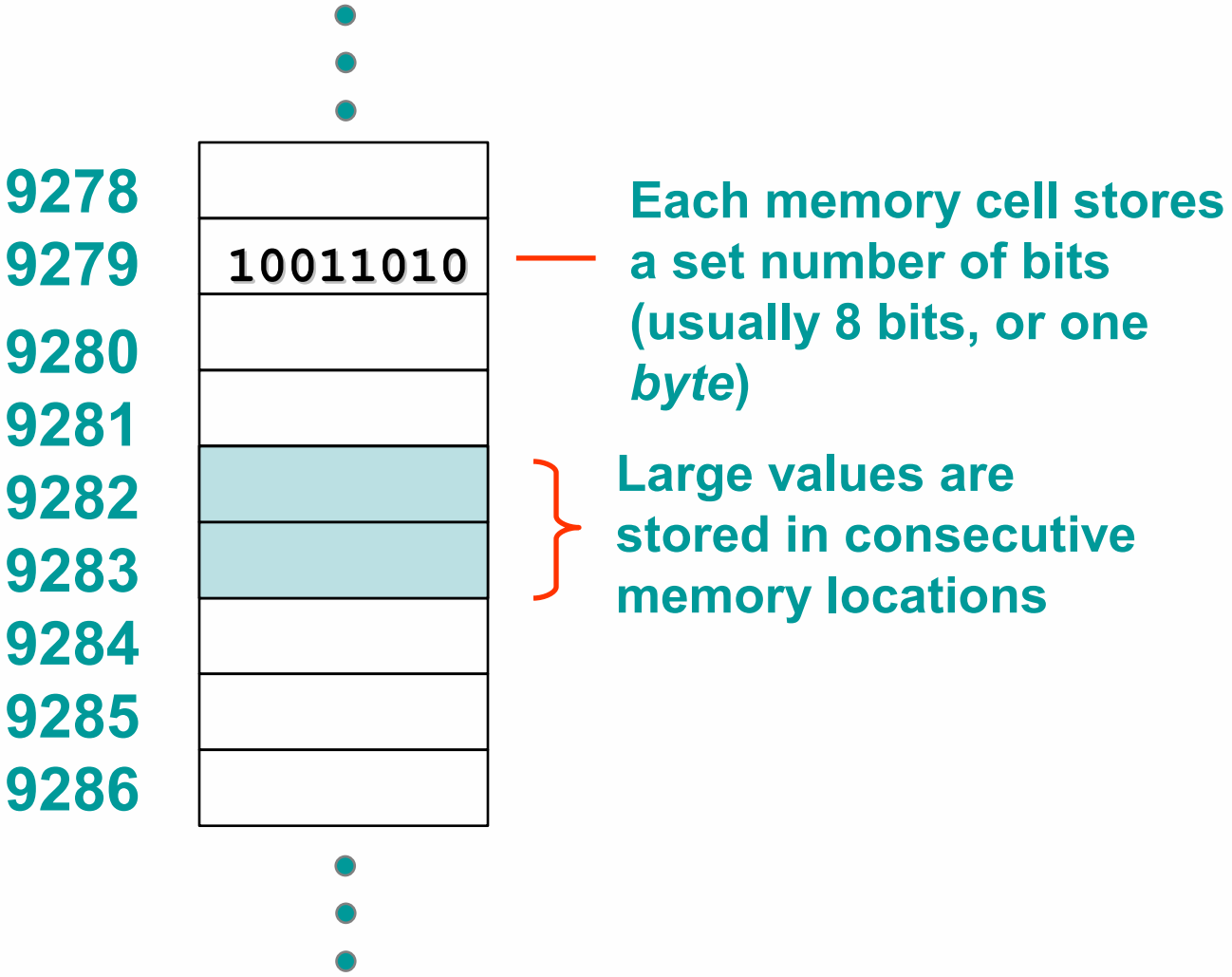
Memory



Main memory is divided into many memory locations (or *cells*)

Each memory cell has a numeric *address*, which uniquely identifies it

Storing Information



Storage Capacity

- **Every memory device has a *storage capacity*, indicating the number of bytes it can hold**
- **Capacities are expressed in various units:**

Unit	Symbol	Number of Bytes
kilobyte	KB	2^{10} = 1024
megabyte	MB	2^{20} (over 1 million)
gigabyte	GB	2^{30} (over 1 billion)
terabyte	TB	2^{40} (over 1 trillion)

Memory

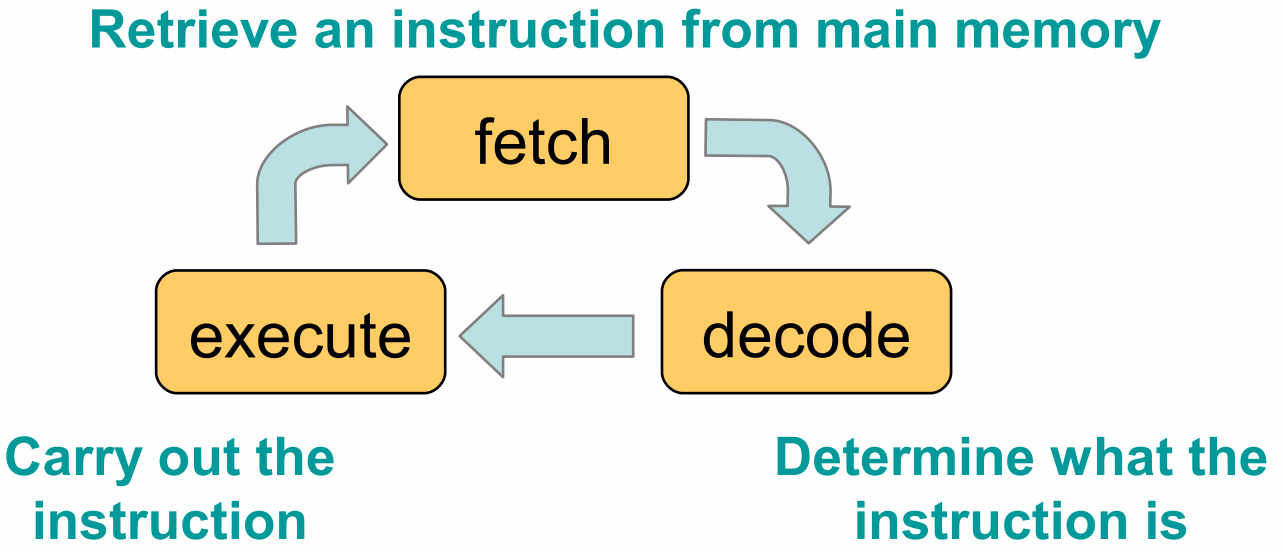
- Main memory is *volatile* - stored information is lost if the electric power is removed
- Secondary memory devices are *nonvolatile*
- Main memory and disks are *direct access* devices - information can be reached directly
- The terms *direct access* and *random access* often are used interchangeably
- A magnetic tape is a *sequential access* device since its data is arranged in a linear order - you must get by the intervening data in order to access other information

RAM vs. ROM

- *RAM* - Random Access Memory (direct access)
- *ROM* - Read-Only Memory
- The terms RAM and main memory are basically interchangeable
- ROM could be a set of memory chips, or a separate device, such as a CD ROM
- Both RAM and ROM are random (direct) access devices!
- RAM probably should be called Read-Write Memory

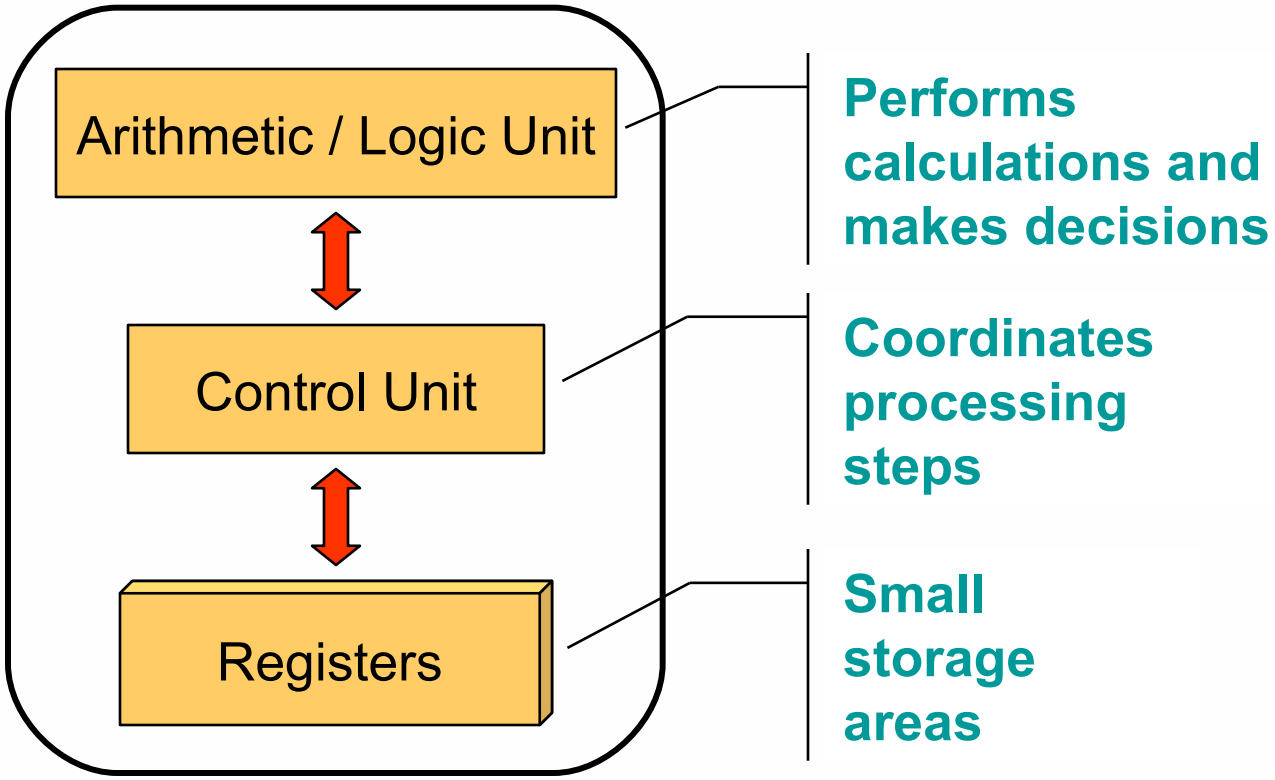
The Central Processing Unit

- **CPU interacts with the main memory to perform all fundamental processing in a computer**
- **CPU is on a chip called a *microprocessor***
- **It continuously follows the *fetch-decode-execute cycle:***



The Central Processing Unit

- **The CPU contains:**



The Central Processing Unit

- The speed of a CPU is controlled by the *system clock*
- The system clock generates an electronic pulse at regular intervals
- The pulses coordinate the activities of the CPU
- The speed is usually measured in *gigahertz* (GHz)

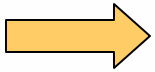
Monitor

- The size of a monitor (17") is measured diagonally, like a television screen
- Most monitors these days have *multimedia* capabilities: text, graphics, video, etc.
- A monitor has a certain maximum *resolution* , indicating the number of picture elements, called *pixels*, that it can display (such as 1280 by 1024)
- High resolution (more pixels) produces sharper pictures

Outline

Computer Processing

Hardware Components



Networks

Programming Languages

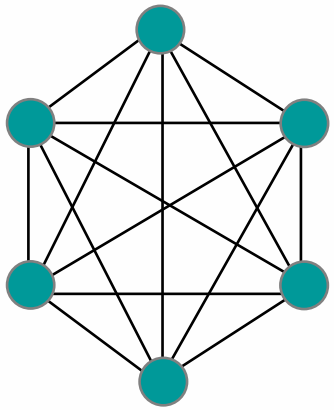
Networks

- A *network* is two or more computers that are connected so that data and resources can be shared
- Most computers are connected to some kind of network
- Each computer has its own *network address*, which uniquely identifies it among the others
- A *file server* is a network computer dedicated to storing programs and data that are shared among network users

Network Connections

- Each computer in a network could be directly connected to every other computer in the network
- These are called *point-to-point* connections

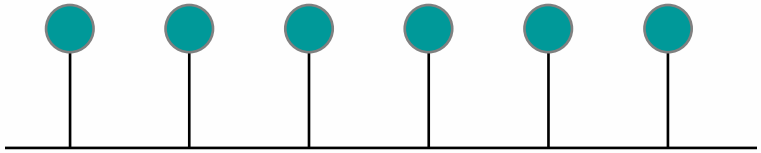
Adding a computer requires a new communication line for each computer already in the network



This technique is not practical for more than a few close machines

Network Connections

- Most networks share a single communication line
- Adding a new computer to the network is relatively easy

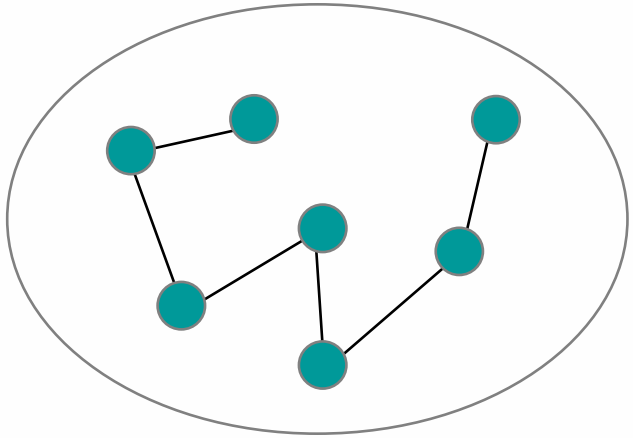


Network traffic must take turns using the line, which introduces delays

Often information is broken down in parts, called *packets*, which are sent to the receiving machine and then reassembled

Local-Area Networks

A *Local-Area Network* (LAN) covers a small distance and a small number of computers

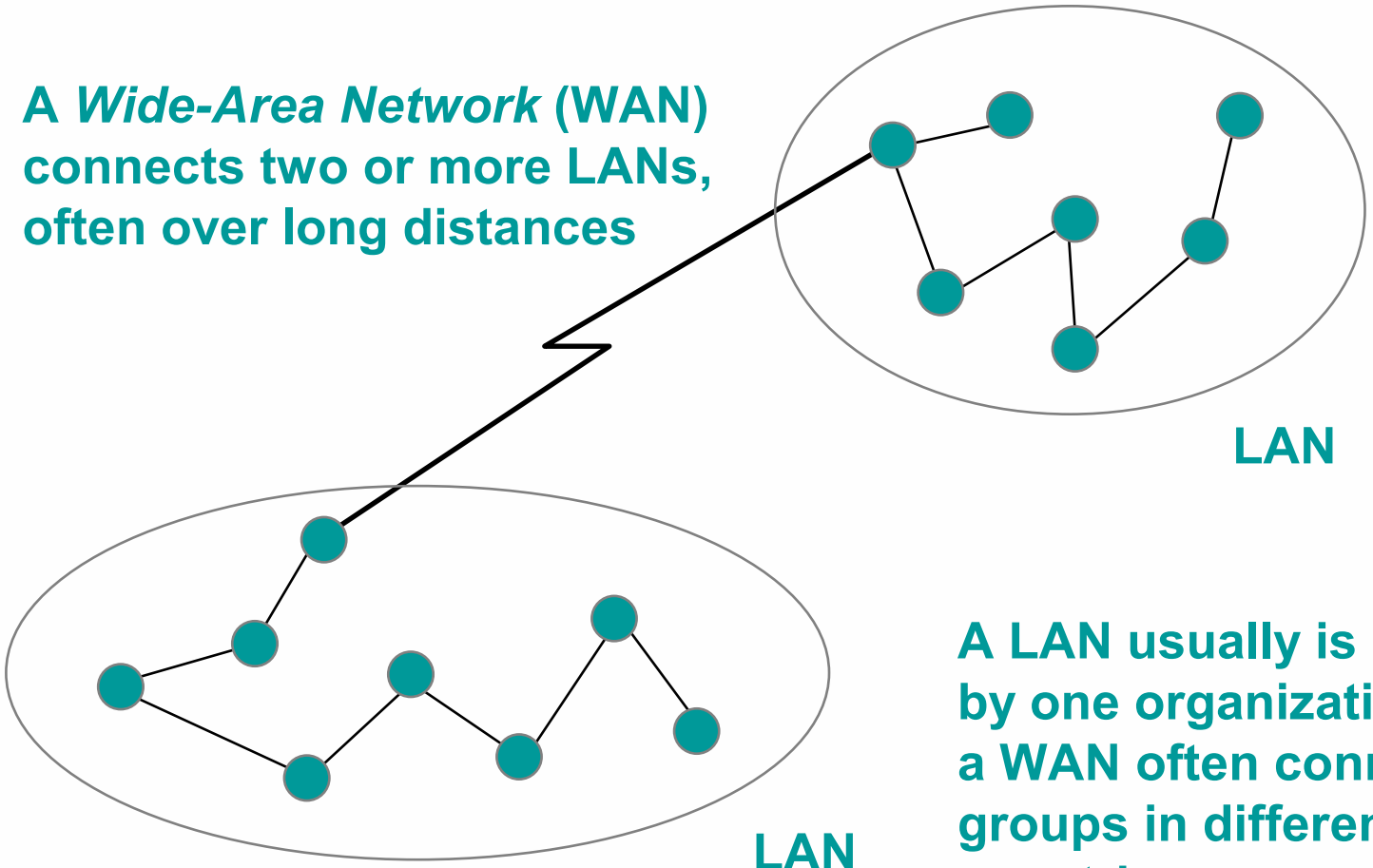


LAN

A LAN often connects the machines in a single room or building

Wide-Area Networks

A *Wide-Area Network* (WAN) connects two or more LANs, often over long distances



A LAN usually is owned by one organization, but a WAN often connects groups in different countries

The Internet

- The *Internet* is a WAN which spans the entire planet
- The word Internet comes from the term *internetworking*
- It started as a United States government project, sponsored by the Advanced Research Projects Agency (ARPA) - originally it was called the ARPANET
- The Internet grew quickly throughout the 1980s and 90s

IP and Internet Addresses

- Each computer on the Internet has a unique *IP address*, such as:

```
204.192.116.2
```

- Most computers also have a unique Internet name, which also is referred to as an *Internet address*:

```
spencer.villanova.edu

kant.gestalt-llc.com
```

- The first part indicates a particular computer (`spencer`)
- The rest is the *domain name*, indicating the organization (`villanova.edu`)

Domain Names

- **The last part of a domain name, called a *top-level domain* (TLD), indicates the type of organization:**

edu - educational institution
com - commercial entity
org - non-profit organization
net - network-based organization

Sometimes the suffix indicates the country:

uk - United Kingdom
au - Australia
ca - Canada
tr - Turkey

New TLDs have recently been added:

biz, info, tv, name

Domain Names

- A domain name can have several parts
- Unique domain names mean that multiple sites can have individual computers with the same local name
- When used, an Internet address is translated to an IP address by software called the *Domain Name System* (DNS)
- There is no one-to-one correspondence between the sections of an IP address and the sections of an Internet address

The World Wide Web

- The *World Wide Web* allows many different types of information to be accessed using a common interface
- A *browser* is a program which accesses and presents information
  - text, graphics, video, sound, audio, executable programs
- A Web document usually contains *links* to other Web documents
- The term Web comes from the fact that information is not organized in a linear fashion

The World Wide Web

- **Web documents are often defined using the *HyperText Markup Language* (HTML)**
- **Information on the Web is found using a *Uniform Resource Locator* (URL):**

```
http://www.lycos.com

http://www.villanova.edu/webinfo/domains.html

ftp://java.sun.com/applets/animation.zip
```

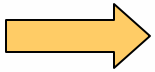
- **A URL indicates a protocol (http), a domain, and possibly specific documents**

Outline

Computer Processing

Hardware Components

Networks



Programming Languages

Problem Solving

- The purpose of writing a program is to solve a problem
- The general steps in problem solving are:
  - Understand the problem
  - Dissect the problem into manageable pieces
  - Design a solution
  - Consider alternatives to the solution and refine it
  - Implement the solution
  - Test the solution and fix any problems that exist

Problem Solving

- Many software projects fail because the developer didn't really understand the problem to be solved
- We must avoid assumptions and clarify ambiguities
- As problems and their solutions become larger, we must organize our development into manageable pieces
- This technique is fundamental to software development

Programming Languages

- A *programming language* specifies the words and symbols that we can use to write a program
- A programming language employs a set of rules that dictate how the words and symbols can be put together to form valid *program statements*

Syntax and Semantics

- The *syntax rules* of a language define how we can put together symbols, reserved words, and identifiers to make a valid program
- The *semantics* of a program statement define what that statement means (its purpose or role in a program)
- A program that is syntactically correct is not necessarily logically (semantically) correct
- A program will always do what we tell it to do, not what we meant to tell it to do

Language Levels

- There are four programming language levels:
  - machine language
  - assembly language
  - high-level language
  - fourth-generation language
- Each type of CPU has its own specific *machine language*
- The other levels were created to make it easier for a human being to read and write programs

Programming Languages

- A program must be translated into machine language before it can be executed on a particular type of CPU
- This can be accomplished in several ways
- A *compiler* is a software tool which translates *source code* into a specific target language
- Often, that target language is the machine language for a particular CPU type

Errors

- A program can have three types of errors
- The compiler will find syntax errors and other basic problems (*compile-time errors*)
  - If compile-time errors exist, an executable version of the program is not created
- A problem can occur during program execution, such as trying to divide by zero, which causes a program to terminate abnormally (*run-time errors*)
- A program may run, but produce incorrect results, perhaps using an incorrect formula (*logical errors*)

Basic Program Development