ACS-1803
Introduction to Information Systems

Instructor: Kerry Augustine

# Introducing the Computer

Lecture Outline 9-1

## Introducing the Computer

- Computer Components and Processing Functions

### Introducing the Computer

- Information processor capable of performing electronically substantial computations including numerous arithmetic or logical operations without intervention by a human operator
- Basic architecture:

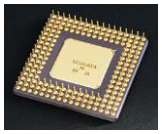
**INPUT** → **CENTRAL PROCESSING UNIT + MAIN MEMORY (*internal*)** → **OUTPUT**

| | | | | | |

**AUXILIARY STORAGE (*external*)**

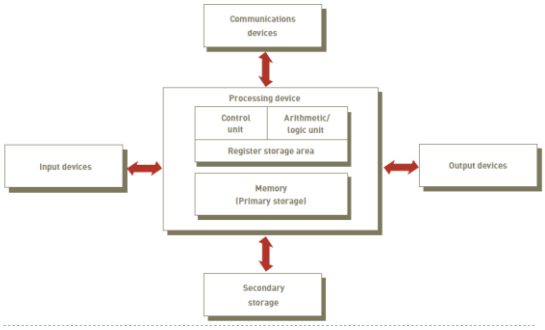
## Computer Components

- Central processing unit (CPU):
  - Arithmetic/logic unit (ALU): Performs mathematical calculations and makes logical comparisons
  - Control unit: Sequentially accesses program instructions, decodes them, and coordinates the flow of data in and out of the ALU, registers, primary storage, and even secondary storage and various output devices
  - Register: Small memory location where instructions to be processed are stored.






## Computer Components (continued)







## Processing Characteristics and Functions

- Clock speed:
  - Series of electronic pulses produced at a predetermined rate that affects machine cycle time
  - Often measured in:
    - Megahertz (MHz): millions of cycles per second
    - Gigahertz (GHz): billions of cycles per second
- Physical characteristics of the CPU
  - Most CPUs are collections of digital circuits imprinted on silicon wafers, or chips, each no bigger than the tip of a pencil eraser

## Memory Characteristics and Functions

- Memory:
  - Provides the CPU with a working storage area for programs and data
  - Rapidly provides data and instructions to the CPU
- Storage capacity:
  - Eight bits together form a *Byte*

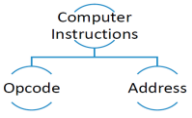



## Main Memory and Instructions

- Cells in main memory hold:
  - instructions and data for the instructions
  - both in electronic form
- Instructions for the CPU tell it to perform sequences of very basic operations
  - e.g., add, subtract, multiply, divide, move, store
  - these are the only kind of instructions that the computer can actually execute
- Every major problem that we want the computer to solve must be broken down into a series of instructions at this simple level




## Sample Machine Level Program




Instruction for a computer: **opcode + address**

- Opcodes (engineers decide on these):
  - 008 - clear accumulator and add to it the contents of the main memory address that follows this opcode
  - 009 - add to the accumulator the contents of the main memory address that follows this opcode
  - 010 - store the result from the accumulator in the main memory address that follows this opcode

example of an instruction: **008 003**

## Machine Level Program – First Generation

**Instruction is : 008 003**

008 - load into accumulator in ALU
003 - **whatever is in address** (cell) **3 in memory**

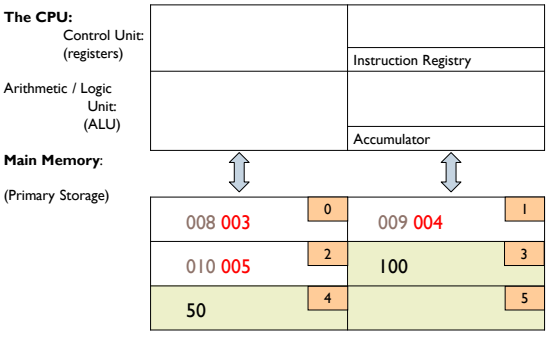
| | | | |
|---|---|---|---|
| **program**: | memory cell | 0: | 008 003 |
| | | 1: | 009 004 |
| | | 2: | 010 005 |
| **data**: | memory cell | 3: | 000 100 |
| | memory cell | 4: | 000 050 |

Instructions are transferred, from memory into the CPU's control unit, one by one, where they are placed in a register and decoded by "wires"



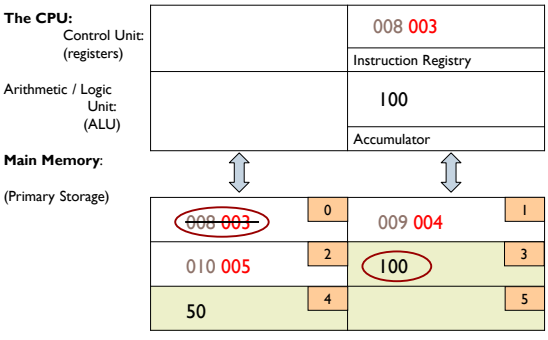
10

## Basic Workings of a CPU:



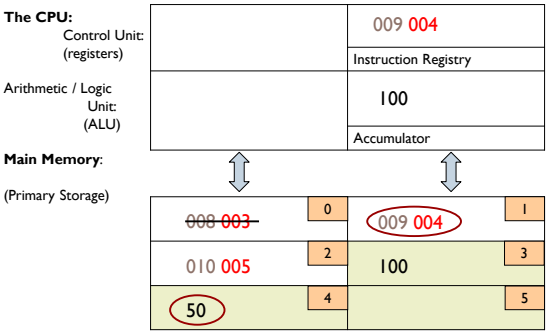

11

## Basic Workings of a CPU:



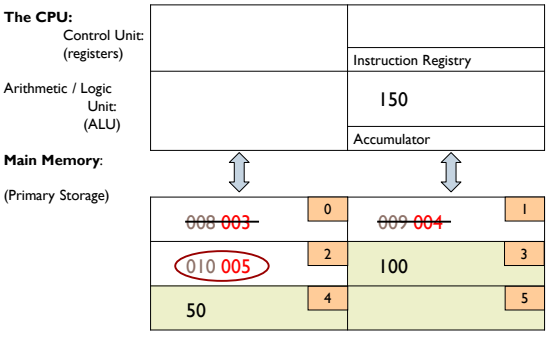

12

## Basic Workings of a CPU:



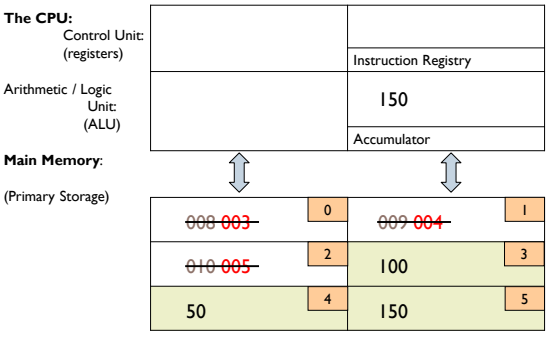

13

## Basic Workings of a CPU:




14

## Basic Workings of a CPU:




15

## How the Computer Understands Instructions

- The machine fetches instructions (from memory), decodes and executes (in CPU) and stores results of the execution (in memory)
  - example of an instruction for CPU: 008 003
- However, such an instruction must be represented electronically, ONLY in terms of + or -
  - 008 003 (base 10)
  - 1000 0011(base 2) +--- --++ (electronic form)
    - This is how the instruction looks in the machine





## How the Computer Understands Instructions

| Base 10 | Base 2 | |
|---|---|---|
| **Decimal pattern** | **Binary numbers** | **Electronic form** |
| 0 | 0 | - |
| 1 | 1 | + |
| 2 | 10 | + - |
| 3 | 11 | + + |
| 4 | 100 | + - - |
| 5 | 101 | + - + |
| 6 | 110 | + + - |
| 7 | 111 | + + + |
| 8 | 1000 | + - - - |
| 9 | 1001 | + - - + |





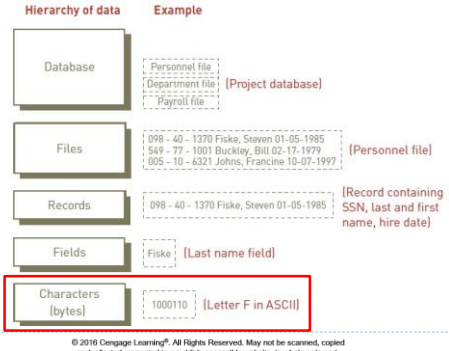
## How the Computer Understands Instructions

- Base 10 → Use 10 different digits to represent numbers
- Base 2 → Use only two digits to represent numbers.

```
  1 1 1 1 1 (carried digits)
  0 1 1 0 1  (13)
+ 1 0 1 1 1  (23)
-------------
= 1 0 0 1 0 0 = 36
```

- Binary is a base-2 system, each digit represents an increasing power of 2, with the rightmost digit representing $2^0$, the next representing $2^1$, then $2^2$, and so on. T
- To determine the decimal representation of a binary number simply take the sum of the products of the binary digits and the powers of 2 which they represent.
- For example, the binary number 100100 is converted to decimal form as follows:
- $= [(1) \times 2^5] + [(0) \times 2^4] + [(0) \times 2^3] + [(1) \times 2^2] + [(0) \times 2^1] + [(0) \times 2^0]$
- $= [1 \times 32] + [0 \times 16] + [0 \times 8] + [1 \times 4] + [0 \times 2] + [0 \times 1]$
- $100100_2 = 36_{10}$

## Hierarchy of Data

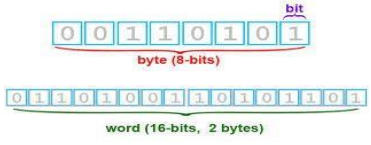





19

## Unit of Digital Measure

- Bit (a binary digit):
  - Circuit that is either on (1) or off (0)
- Byte:
  - Made up of eight (8) bits
- Character:
  - Basic building block of information – two (2) or more bytes






20

## Digital Measure

| Name | Abbreviation | Number of Bytes |
|---|---|---|
| Byte | B | 1 |
| Kilobyte | KB | $2^{10}$ or approximately 1,024 bytes |
| Megabyte | MB | $2^{20}$ or 1,024 kilobytes (about 1 million) |
| Gigabyte | GB | $2^{30}$ or 1,024 megabytes (about 1 billion) |
| Terabyte | TB | $2^{40}$ or 1,024 gigabytes (about 1 trillion) |
| Petabyte | PB | $2^{50}$ or 1,024 terabytes (about 1 quadrillion) |
| Exabyte | EB | $2^{60}$ or 1,024 petabytes (about 1 quintillion) |
| Zettabyte | ZB | $2^{70}$ or 1,024 exabytes (about 1 sextillion) |
| Yottabyte | YB | $2^{80}$ or 1,024 zetabytes (about 1 septillion) |



21

## How the Computer Understands Instructions

- Instructions at this level (+ and -) are said to be in machine language
- Earliest programs were written in machine language (first generation language)
- Then, a coding system was developed
- each character on keyboard is represented by a specific sequence of 0s and 1s
- (ASCII or EBCDIC – agreed upon coding schemes)





## Processing – Language Binary Example

**American Standard Code for Information Interchange (ASCII)**

| Character | ASCII-8 Binary Code | Character | ASCII-8 Binary Code |
|---|---|---|---|
| A | 0100 0001 | S | 0101 0011 |
| B | 0100 0010 | T | 0101 0100 |
| C | 0100 0011 | U | 0101 0101 |
| D | 0100 0100 | V | 0101 0110 |
| E | 0100 0101 | W | 0101 0111 |
| F | 0100 0110 | X | 0101 1000 |
| G | 0100 0111 | Y | 0101 1001 |
| H | 0100 1000 | Z | 0101 1010 |
| I | 0100 1001 | 0 | 0011 0000 |
| J | 0100 1010 | 1 | 0011 0001 |
| K | 0100 1011 | 2 | 0011 0010 |
| L | 0100 1100 | 3 | 0011 0011 |
| M | 0100 1101 | 4 | 0011 0100 |
| N | 0100 1110 | 5 | 0011 0101 |
| O | 0100 1111 | 6 | 0011 0110 |
| P | 0101 0000 | 7 | 0011 0111 |
| Q | 0101 0001 | 8 | 0011 1000 |
| R | 0101 0010 | 9 | 0011 1001 |

**Types of Binary**

**Micro Computers**

- ASCII - 8 bit
- Extended – 8 bit

**Mainframe Computers**

- EBCIDIC – 8 bit
- Extended Binary Coded Decimal Interchange Code

**Other Types**

- Unicode – 16 bit
- Universal Character Set
- Used for international languages



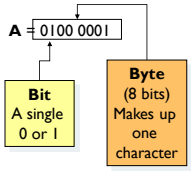


## Processing – Language

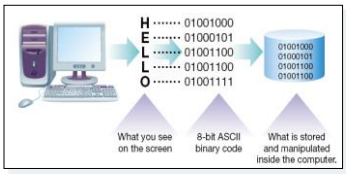
**Binary or Machine Language (First General Language)**

- The language that all computers use
- IT is expressed in 0s or 1s only (see below)
- Binary utilizes Base-2 math to convert from normal characters to binary code (e.g. A = 0100 0001 in binary)

**Binary Example**




**How a Computer Uses it**



## How did a coding system make programming easier?

- Now **programs could be written in <u>symbolic</u> machine language** (assembly language) **because *letters could be entered into a computer* in 0s and 1s**
- How would you write your first name in Binary?




## Assembly Language – Second Generation

e.g.,

```
CLA X
ADD Y
STO Z
X
Y
Z
```

(second generation language)

ADDING TWO NUMERS IN ASSEMBLY LANGUAGE

A translation program [assembler], itself in machine language, would translate this code into actual machine language for the CPU




## Translating Assembly Language

- Programmer writes CLA X
- Machine receives

0100 0011 0100 1100 0100 0001 0101 1001

C L A X

(if there was no ascii we couldn't get this in)

- Assembler program translates this to:

1000 0011 (008 003) [equivalent machine language instruction]

## Higher Level Languages

Assembly language [second generation] - low level:

- **one** statement in assembly language translates into
- **one** statement in machine language

**A complicated, "real world" problem, still had to be broken down into small steps for the CPU**

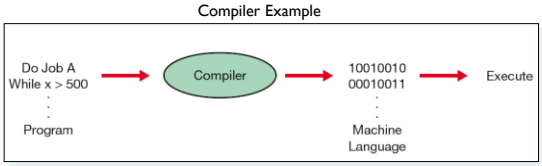
Then came third generation languages (high-level)
- **one** statement in 3GL translates into
- **many** statements in machine language


28

## Compilers and Interpreters

**Compilers**
These highly-specialized software applications are used to convert program instructions (source code) into the machine code (object code) prior to being loaded into a computer's secondary storage

Compiler Example





29

## Third Generation Languages

FORTRAN 3rd Generation Language:

**Z = X + Y**

(will be translated to mach. language by FORTRAN compiler)

COBOL 3rd Generation Language:

**ADD Y TO X GIVING Z.**

(will be translated to mach. language by COBOL compiler)


30

## Third Generation Languages

- Not necessary to think at the level of a machine
- Translation program [*compiler or interpreter*] translates 3GL to machine language

- However, in a 3GL, we still have to tell the computer both **WHAT** to do and **HOW** to do it.
- We call this **PROCEDURAL** Language

- Different 3GLs:

| | |
|---|---|
| **COBOL** (business) | **FORTRAN** (scientific) |
| BASIC | PASCAL |
| C, C++, C# | JAVA |

- each 3GL has different grammar; suited to different problems




## Fourth Generation Languages

- Much more user-friendly

- Tell the computer **WHAT** to do but not **HOW** to do it.
- We call this **NON-PROCEDURAL** Language

- Eg: average <list of numbers>
  - exist only for specific problems / uses

Different 4GLs: DOS
dBASE
SQL
PowerBuilder




## Fourth Generation Computing

- We can also call common application software
  - Word processing
  - Spreadsheets
  - Web browsers
  - Multimedia programs

Fourth generation (non-procedural) software [WHAT to do; not HOW] *but they are* ***not, properly, languages***
- Sometimes called productivity tools

-They use a **GRAPHICAL USER INTERFACE**

## Procedural and Non-procedural Computing

**PROCEDURAL (3rd Generation Language)**

- Need to tell the computer **WHAT** you want **and HOW** to do it (how to *proceed)*

- Need to **have an *algorithm*** for the problem (sequence of logical steps necessary to solve the problem)

-Need to **code the algorithm** in a procedural (3rd Gen) language

**NON-PROCEDURAL (4th Generation Language)**

- Tell the computer what to do, **but not** how to do it.





## Finding the Average of Numbers

- **AVERAGE: 232, 452, 554, 667, 932, 122;**

- **The Algorithm:**
  - NNum = 0; SumNum=0
  - While there are numbers to read
    - Read a number
    - Add 1 to NNum
    - Add the number to SumNum
  - End While
  - Average = SumNum / NNum
  - Print "Average is:", Average





## Coding the Algorithm

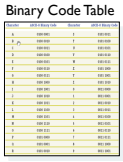
- The algorithm (set of steps) will now be coded in a non-procedural language: Microsoft Excel
- This program tells the computer HOW to find the average
- The program compiles to machine language using an algorithm


- Type in the numbers into a box
- **Click a button for "Average"** (using GUI)

## First to Second Generation Languages

1st GL
Machine Language:
1000 0011
1001 0100
1010 0101

Binary Code Table



Computer Instructions
Opcode Address

2nd GL
Assembly Language:
CLA X
ADD Y
STO Z
X
Y
Z


37

## Second to Third Generation Languages

2nd GL
Assembly Language:
CLA X
ADD Y
STO Z
X
Y
Z

Compiler

Do Job A
While x > 500
.
.
.
Program

10010010
00010011
.
.
.
Machine Language

3rd GL
High Level Language:
**PROCEDURAL** Language


38

## Third to Fourth Generation Languages

3rd GL
High Level Language:
**PROCEDURAL** Language

- **The Algorithm:**
  - NNum = 0; SumNum=0
  - While there are numbers to read
    - Read a number
    - Add 1 to NNum
    - Add the number to SumNum
  - End While
  - Average = SumNum / NNum
  - Print "Average is:", Average

4th GL
High Level Language:
**Non-PROCEDURAL** Language


- Type in the numbers into a box
- **Click a button for "Average"** (using GUI)


39

## Generations of Programming Languages

**Programming Languages**
Used to generate program instructions and have evolved over time making them more powerful, easier to read and write, and more natural language-focused

**Generations of Programming Languages**

| 1940s | 1950s | mid 1950-60s | 1970s | 1990s |
|---|---|---|---|---|
| 1st | 2nd | 3rd | 4th | 5th |
| Machine<br>Binary | Symbolic<br>Use of symbols | High-Level<br>Use English like words for procedures | Outcome Oriented<br>Use outcome focused words | Artificial Intelligence<br>Natural language (spoken English) |

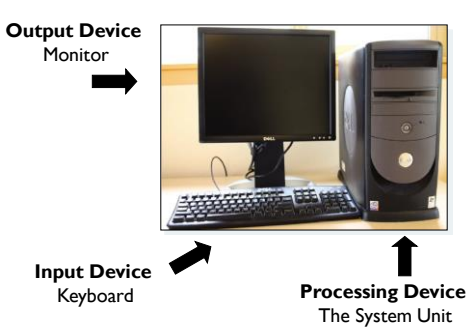




# Computer Hardware

- Microcomputer System

## Microcomputer System



## The Microcomputer

E.g., PC or Apple

- microprocessor (chip) is the CPU
- much elaborate, user-friendly software
- consists of: **system unit (box), monitor (screen), keyboard, mouse, printer**

**In the system unit:**

- **motherboard**, disk drives, CD-ROM drive, cards, cables, power supply





## Motherboard

- Main circuit board: **microprocessor (CPU chip), RAM (Random Access memory - main memory),** buses, cards

- Intel microprocessor chips (past and present):

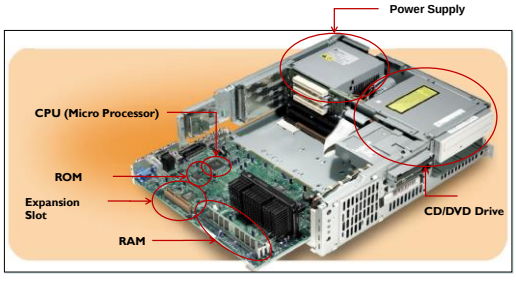
  8088, 8086, 80286, 80386, 80486, Pentium+++

  -speed in MegaHertz (Million of vibrations per second) or GigaHertz (1024 MHz)
  -all processing (calculations) done in the microprocessor





## Processing – Mother Board Example




- A computer's Motherboard holds or connects to all of the computer's electronic components

## Ram/Rom /Expansion Cards

- **RAM**: main memory chips: 2-8 GB +.
  - RAM holds the **Operating System, Application Software, Data**
- **ROM** (Read Only Memory) – burned-in programs to start up the computer
- **Buses** (multi-lane highways) carry instructions from memory to microprocessor and back
- **Expansion Cards**: circuit boards that plug into expansion slots on the motherboard;
  - Links peripheral equipment (printers, disks) with motherboard at the back of the cards are **ports**



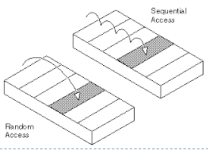

## Computer Storage – Primary/Secondary

- Primary (Internal) Storage:
  - Main memory
  - Stores **instructions** and **data** that are being worked on by the CPU
  - Contents erased when power off
- Secondary (External) Storage:
  - Devices that store large amounts of data, instructions, and information more permanently than allowed with memory
    - Nonvolatility
    - Greater capacity
    - Greater economy
  - Most common forms
    - Magnetic disk, tape
    - Optical storage
    - Solid state




## Secondary Storage Devices

- Sequential access
  - Data must be retrieved in the order in which it is stored
  - Devices used are called sequential access storage devices (SASDs)
- Direct access
  - Records can be retrieved in any order
  - Devices used are called direct access storage devices (DASDs)

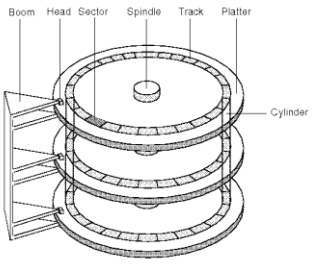


## Magnetic Disk

- hard disk
- platters one below other
- each platter has tracks
- data stored along tracks
- info. picked up by read / write heads

- software and data taken from disk to main memory

- disk allows direct access as opposed to tape which is sequential

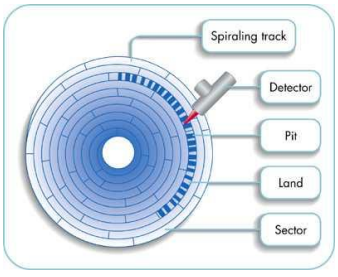





49

## Optical Storage

CD ROM
- laser light instead of magnetic form
- can store much more data in same amount of space
- A CD can hold up to 740 MB Data






50

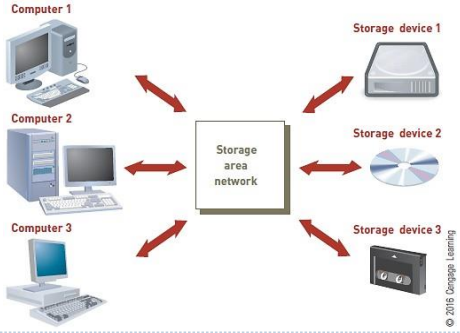
## Secondary Storage Devices

- Digital video disc (DVD):
  - Storage medium used to store software, video games, and movies
- Solid state secondary storage devices:
  - Store data in memory chips rather than magnetic or optical media
  - Have few moving parts, so they are less fragile than hard disk drives
  - High cost per GB of data storage
  - Lower capacity compared to current hard drives



51

## Storage Area Network (SAN)

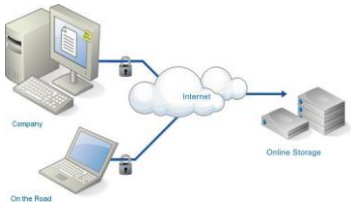





52

## Secondary Storage Devices

- Storage as a Service:
  - Data storage service provider rents space to people and organizations
  - Users access their rented storage space via the Internet






53

## Input Devices

- Devices used to input general types of data:
  - Personal computer input devices
  - Speech recognition technology
  - Digital cameras
  - Touch-sensitive screens
  - Barcode Readers
  - Pen input devices
  - Magnetic stripe card
  - Radio Frequency Identification



54

## Input Devices (Continued)


**Figure 3.6**
RFID Tag
An RFID tag is small compared to current bar-coded labels used to identify items.
(Source: Courtesy of Intermec Technologies Corporation.)




## Output Devices

- Display monitors:
  - Used to display the output from the computer
  - Plasma display:
    - Uses thousands of smart cells (pixels) consisting of electrodes and neon and xenon gases that are electrically turned into plasma to emit light
  - Liquid Crystal Display (LCD):
    - Flat displays that use liquid crystals
  - Light-Emitting Diodes (LEDs):
    - Use a layer of organic material sandwiched between two conductors




## Output Devices

- Printers and plotters:
  - Printers and plotters produce hard copy
  - Laser printers and inkjet printers
  - Multi-function printers
  - 3D printers
  - Plotters are used for general design work

- Digital audio player:
  - Can store, organize, and play digital music files
- E-books:
  - Digital media equivalent of a conventional printed book

## Computer System Types

- Computer systems can range from desktop or portable computers to massive supercomputers
- Two major groups of general-purpose computers
  - Single-user computers with portable and nonportable option
  - Multiple-user computers




## Portable Single-User Computers

- **Handheld computer**: a compact computing device
  - Typically includes a display screen with stylus or touch screen input along with a compact keyboard or numeric keypad
  - Applicable as POS devices
  - Rugged versions are available for military applications
- **Laptop computers** are designed for use by mobile users
  - Notebook and ultrabook computers are smaller than laptop computers
  - Tablet computers are portable, lightweight computers with or without a keyboard




## Multi-User Computer Systems

- A server is employed by many users to perform a specific task, such as running network or Internet applications
- Server systems consist of multiuser computers, including supercomputers, mainframes, and other servers
- Blade server: a server that houses many individual computer motherboards

## Multi-User Computer Systems

- Supercomputers: largest, most powerful, $$$; perform parallel processing
- Mainframes: central, many dumb terminals
- Minicomputers: smaller mainframes
- Microcomputers: can be networked; others: {e.g., portable computers, laptops, tablets, etc.
- Next: Quantum Computing





# Computer Software

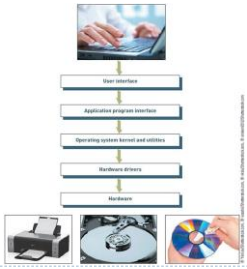
- Operating System and Application Software

## System vs Application Software

- **Systems Software**
  - The set of programs that coordinates the activities and functions of hardware and other programs
  - Each type of systems software is designed for a specific CPU and class of hardware
- **Application Software**
  - Helps users solve particular problems
  - In most cases, resides on the computer's hard disk
  - Can be stored on CDs, DVDs, or USB flash drives

## Operating Systems

- A set of programs that controls computer hardware and acts as an interface with application programs








## Operating Systems Activities

- Controlling common computer hardware functions
- Providing a user interface and input/output management
- Providing a degree of hardware independence
- Managing system memory
- Managing processing tasks
- Providing networking capability
- Controlling access to system resources
- Managing files





## Operating Systems: Processing Tasks

- Five basic task management techniques
  - Multiuser: allows two or more users to run programs at the same time on one computer
  - Multiprocessing: supports running a program on more than one CPU
  - Multitasking: allows more than one program to run concurrently
- Multithreading: allows different threads of a single Program to run concurrently
  - A thread is a set of instructions within an application that s independent of other threads
  - Real time: responds to input instantly

## Current Operating Systems

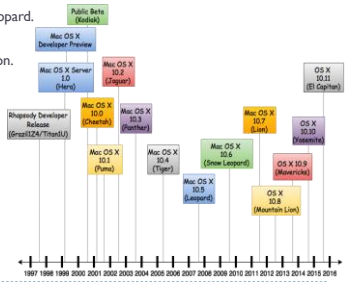
- Microsoft PC operating systems
  - Windows
  - Windows 95
  - Windows XP
  - Windows Vista
  - Windows 7
  - Windows 8
  - Windows 10






## Current Operating Systems

- Apple computer operating systems
  - 4.7 Mac OS X 10.5 Leopard.
  - 4.8 Mac OS X 10.6 Snow Leopard.
  - 4.9 Mac OS X 10.7 Lion.
  - 4.10 OS X 10.8 Mountain Lion.
  - 4.11 OS X 10.9 Mavericks.
  - 4.12 OS X 10.10 Yosemite.
  - 4.13 OS X 10.11 El Capitan.
  - 4.14 macOS 10.12 Sierra.








## Current Operating Systems

- Linux
  - Open-source operating system
  - Red Hat (most common)
- Chrome OS
  - Linux-based operating system designed for netbooks and nettops
  - Designed to run on inexpensive low-power computers
  - Chromium OS: an open-source version of Chrome OS
  - Android-based apps are made available on Chromebooks which makes the platform more general-purpose than a typical thin client.
- Android: an operating system for mobile devices


## Mobile Operating Systems

| Smartphone Operating System | Worldwide Market Share of Sales during 2Q 2013 | Estimated Total Number of Applications Mid-2013 | Estimated Rate of Increase in Number of New Applications |
|---|---|---|---|
| Google Android | 56.5% | >1,000,000 | 800/day |
| Apple iPhone OS | 39.6% | 900,000 | 600/day |
| Microsoft Windows Mobile | 3.3% | 145,000 | 130/day |
| Blackberry Limited, Blackberry | 2.9% | 120,000 | NA |





## Operating Systems - Utilities

**Utilities:** Programs that manage **computer resources** and **files** and may be included in the **operating system** or purchased **separately** as needed

| Utility | Description |
|---|---|
| Backup | Archives files from the hard disk to a diskette or to tapes |
| File defragmentation | Converts a fragmented file stored on your hard disk (one not stored contiguously) into one that will load and be manipulated more rapidly |
| Disk and data recovery | Allows the recovery of damaged or erased information from hard and floppy disks |
| Data compression | Compresses data by substituting a short code for frequently repeated patterns of data, much like the machine shorthand used by court reporters, allowing more data to be stored on a disk |
| File conversion | Translates a file from one format to another, so it can be used by an application other than the one used to create it |
| Antivirus | Monitors and removes viruses—lines of code designed to disrupt the computer's operation and make your life miserable |
| Device drivers | Allows new hardware added to your computer system, such as a game controller, printer, scanner, and so on, to function with your operating system |
| Spam blockers | Monitors your incoming e-mail messages and filters or blocks the message from arriving |
| Spyware detection and removal | Monitors and removes spyware from your computer (see Chapters 4 and 9) |
| Media players | Allows music in formats such as MP3, WMA, or WAV or video in formats such as MPEG, AVI, ASF to be listened to or watched on a computer |





## Application Software

- Application programs:
- Interact with systems software and the systems software directs computer hardware to perform necessary tasks
- Help you perform common tasks, such as:
  - Creating and formatting text documents
  - Performing calculations
  - Managing information
  - Some applications are more specialized

## Application Software

- Proprietary software:
  - One-of-a-kind program for a specific application, usually developed and owned by a single company
- Off-the-shelf software:
  - Existing software program that is purchased
  - Application service provider (ASP):
    - Company that can provide software, support, and computer hardware on which to run the software from the user's facilities over a network




## Proprietary Software Advantages and Disadvantages

| Advantages | Disadvantages |
|---|---|
| You can get exactly what you need in terms of features, reports, and so on. | It can take a long time and significant resources to develop required features. |
| Being involved in the development offers control over the results. | In-house system development staff may be hard pressed to provide the required level of ongoing support and maintenance because of pressure to move on to other new projects. |
| You can modify features that you might need to counteract an initiative by competitors or to meet new supplier or customer demands. | The features and performance of software that has yet to be developed presents more potential risk. |




## Off-the-Shelf Software Advantages and Disadvantages

| Advantages | Disadvantages |
|---|---|
| The initial cost is lower because the software firm can spread the development costs over many customers. | An organization might have to pay for features that are not required and never used. |
| The software is likely to meet the basic business needs—you can analyze existing features and the performance of the pack- age before purchasing. | The software might lack important features, thus requiring future modification or customization. This lack can be very expensive because users must adopt future releases of the soft- ware as well. |
| The package is likely to be of high quality because many customer firms have tested the software and helped identify its bugs. | The software might not match current work processes and data standards. |

## Application Software: Software as a Service (SaaS) and Cloud Computing

- Software as a service (SaaS):
  - Allows businesses to subscribe to Web-delivered business application software by paying a monthly service charge or a per-use fee
  - Can reduce expenses by sharing its running applications among many businesses
- Cloud computing:
  - Use of computing resources, including software and data storage, on the Internet (the cloud) rather than on local computers