W1005
Intro to CS and Programming in MATLAB

Brief History of Computing

Fall 2014
Instructor: Ilia Vovsha

http://www.cs.columbia.edu/~vovsha/w1005

Computer Philosophy

- Computer is a (electronic digital) device for performing computation
- Requires a list of instructions (program) written in special language (programming language)
- Computers can't think (despite the popular sci-fi myth)
- "If you tell me precisely what is that a machine cannot do, then I can always make a machine which will do just that" – John Von Neumann (1948)

Milestones

- 14th Century: Abacus
- 1837: analytical engine description (Charles Babbage)
- Before 1935: person performs computations
- 1936: principles of modern computers (Turing Machines, Alan Turing)
- 1935 – 1945: first digital machines (Iowa State, “Colossus”)
- 1946: Electronic-Numeric-Integrator-and Computer (ENIAC), first electronic large scale, general-purpose computer (UPENN)
- 1946: Von Neumann architecture (stored-program computer, basis of modern architecture)

Milestones

- 1951: the advent of microprogramming
- 1960-70s: early supercomputers, integrated circuits, microprocessors, mainframes and time-sharing
- 1980s: personal computers, hardware cost reduction

Hardware Components

1. Memory (main, secondary, storage media)
2. Central Processing Unit (CPU)
3. Input devices (mouse, keyboard)
4. Output devices (monitor, printer)
5. Network connection

Anatomy of Memory

- Computer memory is a sequence of storage locations (*cells*)
  - Each cell has a unique address (position in memory)
  - Cell contents store data or program instructions
  - A cell is a group of *bytes* (single characters)
  - Bytes are composed from groups of binary digits (*bits* = 0/1)
  - Retrieve and store values from/to cell using electronic signal

Main & Secondary Memory

- Two types of main memory:
  - *Random Access Memory (RAM)*: temporary, volatile, fast, limited but can be increased
  - *Read Only Memory (ROM)*: permanent, can't write/store info, fixed amount
- Secondary:
  - Semi-permanent data storage through tapes/disks
  - Disks are organized into files

CPU

- Two roles:
  1. Coordinate operations
  2. Arithmetic/logical operations on data
- Typical operations:
  - Retrieve instructions
  - Decode instructions
  - Process data
  - Store results
- CPU memory: high-speed locations called *registers*

Networks

- Connect multiple devices to share resources
- *Local Area Network (LAN):* large computer (*server*) shares resources between many computers
- *Wide Area Network (WAN):* network over large geographic area
- *World Wide Web (WWW):* developed by CERN, (1989), effective and uniform system of accessing information on the internet
- *Web browser:* program with a user interface to navigate the web

Software

- Operating System:
  - Controls interaction between user and hardware/software
  - Manages resources
  - Stored in ROM and partially loaded (booting) into RAM
  - Command line (UNIX) or graphical (Windows, Mac OS) interface
- Application Programs:
  - Accomplish specific tasks
  - Must be compatible with OS and hardware

Programming Languages

- Language for writing instructions to machines
- *Machine* language: binary instructions, explicit, low-level operations
- *Assembly* language: user-friendly representations of machine instructions
- *3rd generation* languages: higher-level notation (Fortran, Lisp)
- *Object oriented* languages: abstraction (C++, Java)
- *Scripting* languages: highest level, gluing together computations, shorter programs (Perl, Python, MATLAB)
- High-level language:
  - User-friendly, easy to use, built-in functions (+)
  - Slower, less control (-)

Compilers

- *Compiler*: program that translates code into machine language
  - Reads input source file (*lexical analysis*)
  - Scans file to check for syntax errors i.e. proper grammar (*syntax analysis*)
  - Checks file for semantic consistency (*semantic analysis*)
  - Converts code to machine language (*code generator*)
  - Optimizes code (*code optimizer*)
- Large programs are often compiled in pieces
  - Target machine code can be linked to other files/libraries (*linker*)
  - Puts together all of the files (*loader*) → ready for execution
- Executing a program:
  - CPU examines each program instruction in memory and sends out signals to carry it out

Software Development Method

1. Specify the problem
2. Analyze the problem
3. Design algorithm
4. Implement algorithm
5. Test & Verify code
6. Maintain & Update software