Software

- Applications Software
- Systems Software
  - Operating Systems
  - Utilities / Drivers
  - Compilers / Interpreters
  - IDEs

2 Basic Categories of Software:

- Applications software
  - This is what we usually think of…
  - Programs for end-users
  - Examples: MS office, database programs, word processors, games, instant messaging, email, browsers… and etc..
- Systems software
  - They bridge between applications software and computer hardware
  - Interact with the hardware at a low-level (very basic level)
  - Provides a high-level environment in which we can run applications on
  - Examples: operating systems, compilers, assemblers… etc

- Systems software bridges the gap between hardware and applications software

Image: © Michele Rousseau

Imagine a computer without an OS

- The earliest computers were this way… Remember the ENIAC?
- Here's how it was programmed.
- These women were the operating system

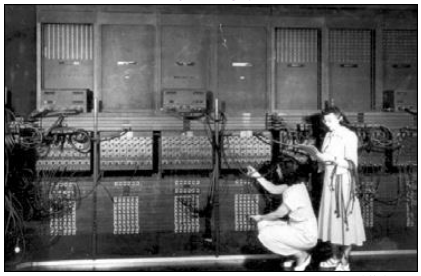


Image: Public Domain
U.S. Army Photo
http://ftp.arl.mil/ftp/historic-computers/

ENIAC 1946

(Electronic Numerical Integrator and Computer)

"The procedure for instructing the ENIAC in its routine, then, consists of setting program switches on the units so that, when stimulated by a program input pulse, the program controls will cause the units to carry out a set of specific operations."

- Adele Goldstine, 1946

Programmers Betty Jean Jennings (left) and Fran Bilas (right) operate the ENIAC's main control panel at the Moore School of Electrical Engineering.

(U.S. Army photo from the archives of the ARL Technical Library)

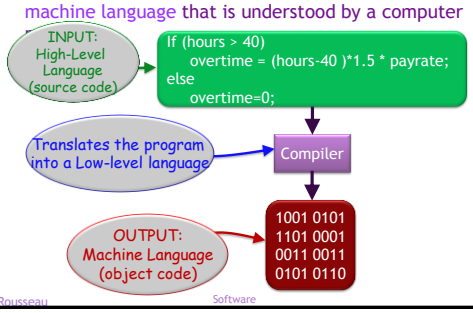
Image: Public Domain
U.S. Army Photo
http://ftp.arl.mil/ftp/historic-com

Operating Systems, Utilities, Drivers

- Operating Systems
  - Manage the systems hardware resources
    - Organizing files (folders etc)
    - Manage downloading
    - Scheduling programs on the processor
    - Interfacing with the internet
    - Managing communication with peripherals
    - Providing security
    - …and lots more (GUI, windows, …)
- Utilities
  - Specialized software (many are built into the O.S.)
  - Enhances the computer's operation
    - backup programs, file compression, etc
  - Safeguards data
    - anti-virus or malware programs
- Drivers
  - Specialized for devices we hook up to our computer
    - Allows for communication of external devices
    - Many of them come with the O.S., but not all of them
    - e.g. Printers, scanners, hard drives

Compilers

- Software that translates high-level programming languages (such as c++, java, etc..) into the low-level machine language that is understood by a computer



Compilers

- high level programming language must be automatically translatable to low-level code
- this puts constraints on the expressiveness of computer programming languages
  - Natural Language is Vague → Computers don't do well with non-precise instructions
- programming languages have to be very precise

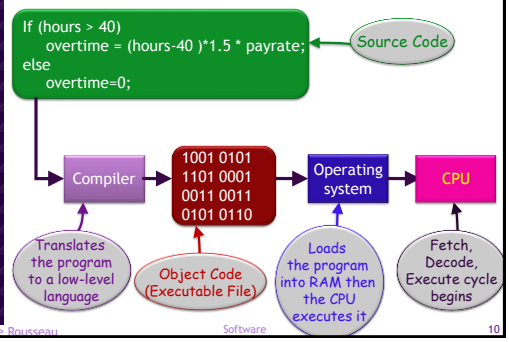
Putting it all together

1. an application is written in a high-level programming language (e.g. c++)
2. the code is translated to machine language →by a compiler
3. when you want to run the application, the operating system loads the code into RAM (random access memory)
4. the fetch/decode/execute cycle is performed

....let's look at an example

Putting it all together

- Summary



Low-Level Languages

- Machine Language
  - Binary code
- Assembly Language
  - Uses abbreviations called opcode
- Assembler
  - Translates assembly code into machine code (binary)

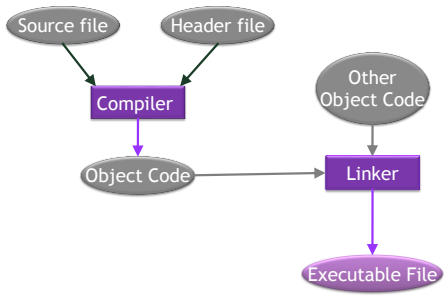
Translating code

- Compilers
  - Converts entire program (source code) into machine language code (object code)
  - Both the source code and object code are stored on the disk
  - We execute (or run) the object code
    - Object code is loaded into memory (RAM)
  - Machine language is processor specific
- Interpreter
  - Translates one sentence at a time (HTML)
  - Source program is interpreted every time!
  - No object code is produced

Our Process

- We use an editor to type in commands
  - We will use eclipse
- Source code
  - a C++ program
- Compiler
  - Translates the source code to binary (provided there are no errors - it will check for errors)
- Object code
  - A C++ program in binary code
- Linker
  - Links your code with libraries → turns your object code into executable code (we see this as a .exe file)
- Libraries
  - Predefined code that we include in a header file

Process for compiling C++



Software Development Tools

- Software development tools
  - Support the process of software development
  - For example: IDEs (integrated development environments)