

Software

CS1A

- * Systems Software
- * Applications Software

(c) Michele Rousseau

Software

1

2 Basic Categories of Software:

- o 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..
- o 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

(c) Michele Rousseau

Software

2

- o Systems software bridges the gap between hardware and applications software



Image: © Michele Rousseau

(c) Michele Rousseau

Software

3

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

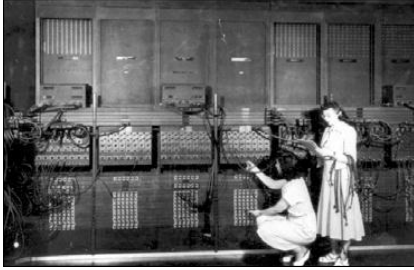


Image: Public Domain
U.S. Army Photo
<http://ftp.arl.mil/ftp/historic-computers/>

4

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)

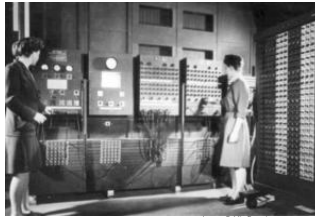


Image: Public Domain
U.S. Army Photo
<http://ftp.arl.mil/ftp/historic-computers/>

5

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

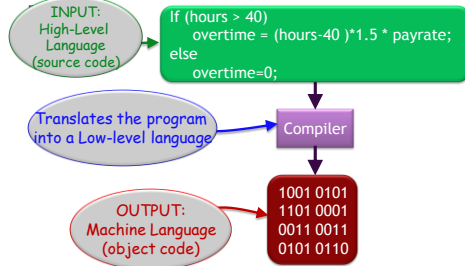
© Michel Rousseau

Software

6

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



M. Michèle Rousseau

Software

7

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

M. Michèle Rousseau

Software

8

Putting it all together

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

....let's look at an example

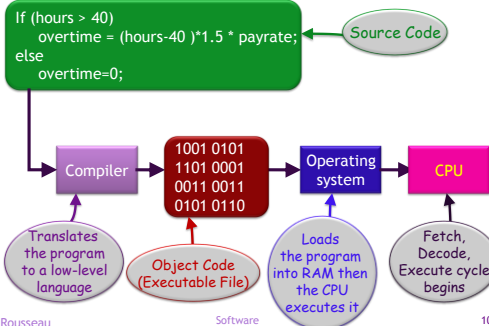
M. Michèle Rousseau

Software

9

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

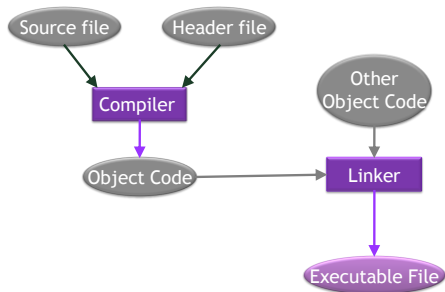
- We use an **editor** to type in commands
 - We will use **eclipse**
- Source code
 - a C++ program
- Compiler
 - Translates the source code to binary
(provided there are no errors - it will check for errors)
- Object code
 - A C++ program in binary code
- Linker
 - Links your code with libraries →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

M. Michèle Rousseau

Software

13

Process for compiling C++



M. Michèle Rousseau

Software

14

Software Development Tools

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

M. Michèle Rousseau

Intro to CS

15
