Laboratorio di Calcolo II

- Introduzione al computing
- UNIX/Linux
- C++
- Lezioni “teoriche” tutti insieme
- Esercitazioni individuali al computer

Docente & Esami

Docente: Fabrizio Bianchi
Tel.: 011/6707331 email: bianchi@to.infn.it
URL: www.to.infn.it/~bianchi/bianchi.html
Esami:
27/6, 25/7, 5/9, 26/9

Textbook & Slides

Suggested Textbook:
C++ How To Program, 4 E, Deitel & Deitel, Prentice Hall
PowerPoint Slides available in:
www.to.infn.it/~bianchi/LaboratorioCalcoloI/Lezzy.ppt

What is Programming?

A sequence of statements that instruct a computer in how to solve a problem is called a program.
The act of designing, writing and maintaining a program is called programming.
People who write programs are called programmers.

What kinds of statements do computers understand?

A computer only understands machine language statements.
A machine language statement is a sequence of ones and zeros that cause the computer to perform a particular action, such as add, subtract, multiply, ...

Machine Language (ML)

ML statements are stored in a computer’s memory, which is a sequence of switches.
For convenience of representation, an “on” switch is represented by 1, and an “off” switch is represented by 0.
ML thus appears to be binary (base-2):
00101110101101010010111010110101
Early Computers

... required a programmer to write in ML...
- Easy to make mistakes!
- Such mistakes are hard to find!
- Not portable -- only runs on one kind of machine!

Programming was very difficult!

A Bright Idea

Devise a set of abbreviations (mnemonics) corresponding to the ML statements, plus a program to translate them into ML.

ADD

Assembler

0010111010101

The abbreviations are an assembly language, and the program is called an assembler.

Assembly Languages

Allowed a programmer to use mnemonics, which were more natural than binary.
+ Much easier to read programs
+ Much easier to find and fix mistakes
- Still not portable to different machines

High Level Languages

Devise a set of statements that are close to human language (if, while, do, ...), plus a program to translate them into ML.

The set of statements is called a high level language (HLL) and the program is called a compiler.

HLL Compilers

Where an assembler translates one mnemonic into one ML statement, a HLL compiler translates one HLL statement into multiple ML statements.

\[ z = x + y; \]

HLLs

High level languages (like C++) are
+ Much easier to read programs
+ Much easier to find and fix mistakes
+ Portable from one machine to another (so long as they keep to the language standard).
Objectives in Programming

A program should solve a problem:
+ correctly (it actually solves the problem)
+ efficiently (without wasting time or space)
+ readably (understandable by another person)
+ in a user-friendly fashion
  (in a way that is easy for its user to use).

Summary

There are "levels" to computer languages:
- ML consists of "low" level binary statements,
  that is hard to read, write, and not portable.
- Assembly uses "medium" level mnemonics:
  easier to read/write, but not portable.
- C++ is a "high" level language that is even
easier to read/write, and portable.