

Les bases de l'informatique : les langages

A. Comment parler au processeur ?

Un processeur parle le **langage machine** :

- C'est une suite de bits interprétée par le processeur.
- C'est le seul langage que le processeur puisse traiter.

Il est composé d'**instructions** et de **données** à traiter codées en binaire.

Exemple : l'instruction 10110000 01100001 met la valeur hexadécimale 61 dans le registre AL. Ici, 10110000 correspond à l'**opcode** et 01100001 à la **donnée**.

Un processeur possède nativement un **jeu d'instructions** composé d'**opcode** (code opération) et d'opérandes. Il existe des opcodes pour faire des opérations arithmétiques, logiques, etc ...
Chaque instruction nécessite un certain temps (généralement un nombre de cycles d'horloge) pour s'exécuter.

B. Comment peut-on parler au processeur ?

Le langage informatique le plus proche du langage machine est l'**assembleur** :

- C'est un langage de bas niveau qui représente le langage machine sous une forme lisible par un humain.
- Les combinaisons de bits du langage machine sont représentées par des **mnémoniques** (des symboles) faciles à retenir : ADD pour l'addition, MOV pour la copie de valeurs, etc

...

Exemple :

L'instruction binaire 10110000 01100001 sera movb \$0x61,%al en assembleur.

Il est en théorie possible de désassembler le code machine d'un programme. En pratique, c'est plus compliqué que cela car il est parfois difficile de savoir si une valeur en mémoire est un opcode ou une donnée. Dans tous les cas, on ne pourra jamais remonter plus haut que l'assembleur.

C. Comment faire pour parler au processeur ?

Les contraintes sont les suivantes :

- L'ordinateur ne sait exécuter qu'un nombre limité d'opérations élémentaires représentées par des instructions codées en binaire.
- L'ordinateur ne comprend que le langage machine.

On est donc obligé d'utiliser des outils pour traduire un langage simple vers le langage machine.

On a ensuite créé des langages de programmation évolués (le plus proche possible de l'humain) et utilisables sur n'importe quel ordinateur.

L'arrivée des premiers langages évolués comme le Cobol et le Fortran datent des *années 1950*.

D. Les langages de programmation

Quel que soit le langage évolué, il faut le **traduire en langage machine**. Il existe des approches différentes :

- **Assemblage** (assembleur) : traduction du mnémonique (symbole) en opcode.
- **Compilation** (Pascal, C, C++, ...) : traduction de l'ensemble du programme écrit en langage évolué (**code source**) en langage machine (**code objet**). Le programme en langage machine sera ensuite exécuté.
- **Interprétation** (Javascript, PHP, Python, ...) : traduction de chaque instruction avant de l'exécuter. On nomme souvent ces programmes des **scripts**.
- **Compilation et Interprétation** (Java) : compilation en un code intermédiaire (bytecode) qui n'est pas du code machine mais un code pour une machine virtuelle . Ce code est ensuite interprété par un interpréteur (la machine virtuelle).

Les langages n'ont pas les mêmes performances. On utilise pas les mêmes langages pour faire un OS, un tableur, un site web ou une application pour smartphone.

H. Les types scalaires

Une **variable scalaire** est destinée par son **type** à contenir une **valeur** atomique.

Les valeurs atomiques sont :

- les **booléens**
- les **nombre entiers**
- les **nombre à virgule flottante**

Remarque : les *caractères* sont en fait des nombre entiers ! En effet, on est obligé de convertir les caractères sous une forme binaire qui constitue un code. Par exemple, le code ASCII du caractère 'A' est 65 (0x41 en **hexadécimal**).