

CONSTRUIS TON PROPRE COMPILATEUR, LE LANGAGE MACHINE


SIMPLETRON MACHINE LANGUAGE

Par Joel Sandé

LE LANGUAGE MACHINE

- Le **langage machine**, ou code **machine**, est la suite de bits qui est interprétée par le processeur d'un ordinateur exécutant un **programme** informatique. C'est le **langage** natif d'un processeur, c'est-à-dire le seul qu'il puisse traiter. Il est composé d'instructions et de données à traiter codées en binaire.

(selon wikipédia)



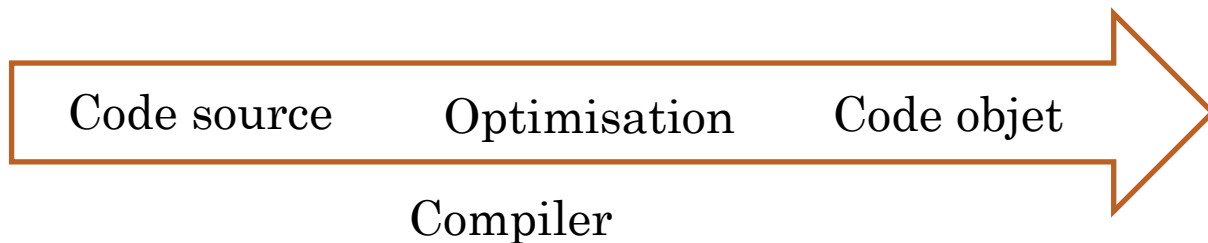
UN COMPILATEUR

- Un compilateur est un langage de programmation, ou un ensemble de programmes qui transforme le code source d'un programme (haut niveau) en bas niveau compréhensible par l'ordinateur.
- Le langage Assembleur est un langage de bas niveau.
- Le langage Java est en langage de haut niveau
- Un compilateur opère des opérations telles que :
 - l'analyse lexical, - l'analyse sémantique, -le parsing, - traduction de la synthaxe, génération du code, ...



UN COMPILATEUR

- Il convertit les instructions lisibles par l'humain en instructions lisibles par l'ordinateur.
- Il prend en générale plus de temps qu'un interpréteur parce qu'il fait du rearrangement et de l'optimisation de tout le code source à la fois (peut avoir une meilleure façon de structurer le code).



UN INTERPRÉTEUR

- L'interpréteur convertit les instructions du langage humain CHAQUE FOIS que le programme est exécuté. Il interprète chaque instruction et produit les résultats. Il est plus rapide qu'un compilateur.
- C'est quoi la différence ? Ils font la même chose, mais différemment. Le compilateur le fait une fois pour toute, et l'interpréteur le fait chaque fois que le programme est exécuté, ligne par ligne.



LANGUAGES COMPILÉ VS INTERPRÉTÉS

- Voici une liste de langages compilés :
C#, C++, Java, C, Pascal, Fortran, Cobol.
- Voici une liste de langages interprétés :
Basic, Perl, Python, Rugby, Mathematica, Matlab, PHP, Basic, Prolog.

Voici un vidéo pour mieux comprendre le concepte

<https://www.youtube.com/watch?v=1veGrAjgJBY>

Une vidéo complémentaire

<https://www.youtube.com/watch?v=1OukpDfsuXE>



CRÉONS NOTRE PROPRE ORDINATEUR : LE SIMPLETRON

- Créons un ordinateur qui s'appelle **le simpletron**. C'est une simple machine, mais puissante. Le simpletron exécute un programme écrit en **Simpletron Machine Language (SML)** ou **Langage Simpletronique (LS)**.
- Il contient un **Accumulateur**, un registre dans lequel une information est introduite avant que le Simpletron l'utilise pour faire **un calcul ou l'examiner**; C'est un registre dans lequel des **opérations arithmétiques** peuvent être effectuées.
- Un **espace de stockage** qui contient les **instructions à exécuter** et les informations à retenir

CRÉONS NOTRE PROPRE ORDINATEUR : LE SIMPLETRON

- Chacune des informations dans le simpletron est recueillie en terme de **mots**. Un mot est une suite digitale de 5 chiffres. et d'un signe, comme **+21013**, **+32009**, **-00022**, **+00005**, etc. Lorsqu'un mot représente une instruction, son signe est toujours positif. Lorsqu'il représente une information, son signe peut être positif ou négatif.
- Avant d'exécuter un programme SML, vous devez placer du code en memoire. La premiere instruction est toujour placée en position 00. La mémoire de l'ordinateur simpletronique compte jusqu'à 100 mots numérotés par leur position : 00, 01, ..., 99.



CRÉONS NOTRE PROPRE ORDINATEUR : LE SIMPLETRON

- Pour chaque mot qui représente une instruction.
- Les trois premiers caractères indiquent le code d'opération.
- Les deux derniers caractères d'un mot représentent l'adresse mémoire (la position) qui contient l'opérande auquel l'opération s'applique.



INSTRUCTIONS

110	Addition
120	Soustraction
130	Multiplication
140	Division
210	Lecture d'un mot à partir du clavier et enregistrement à une position spécifique
220	Affichage d'un mot à l'écran à partir d'une position spécifique
310	Chargement d'un mot dans l'accumulateur à partir d'une position spécifique
320	Enregistrement d'un mot à une position spécifique à partir de l'accumulateur
410	Branchement vers une position spécifique
420	Branchement vers une position spécifique si l'accumulateur est négatif
430	Branchement vers une position spécifique si l'accumulateur est nul
440	Fin



TRAVAUX DIRIGÉS

- Exemple 1 : Nous allons écrire un programme LS qui lit deux nombres au clavier, les additionne puis affiche le résultat à l'écran.
- Exemple 2 : Nous allons écrire un programme LS qui lit deux nombres au clavier puis détermine le plus grand des deux.



EXEMPLE 1

Position	Mot	Signification
00	+21007	Lecture de la variable A qui est stockée en position 07
01	+21008	Lecture de la variable B qui est stockée en position 08
02	+31007	Chargement de la variable A dans l'accumulateur
03	+11008	Addition de B
04	+32009	Enregistrement du résultat dans la variable C, stockée en position 09
05	+22009	Affichage à l'écran de la variable C
06	+44000	Fin
07	? ?????	Variable A
08	? ?????	Variable B
09	? ?????	Variable C



EXEMPLE 2

Position	Mot	Signification
00	+21009	Lecture de la variable A
01	+21010	Lecture de la variable B
02	+31009	Chargement de la variable A dans l'accumulateur
03	+12010	Soustraction de B (A-B)
04	+42007	Si le résultat dans l'accumulateur est négatif, va à l'instruction 07
05	+22009	Affichage à l'écran de la variable A
06	+44000	Fin
07	+22010	Affichage à l'écran de la variable B
08	+44000	Fin
09	? ?????	Variable A
10	? ?????	Variable B

Remarquez : l'instruction en position 04 qui effectue un branchement vers une autre instruction lorsqu'une condition est rencontrée, un peu comme un if.



ASTUCES

- Notez que pour faire fonctionner un programme, l'ordinateur simpletronique peut initialiser des variables (stocker une valeur dans une case mémoire) avant de lancer la première instruction. Il pourra ainsi utiliser une position dans l'espace de stockage comme compteur, comme incrément, comme valeur limite, etc.



ASTUCES

- Suggestions pour réaliser les exercices sur papier et faciliter la programmation qui suivra :
 - Avoir sous la main une feuille de papier pour griffonner ce qui se passe dans la tête de l'ordinateur
 - Dessiner un tableau pour représenter l'espace de stockage
 - Réserver les positions à la fin pour placer les variables (on ne sait pas au départ combien d'instructions il y aura avant d'arriver à l'instruction de fin)
 - Dessiner une case à part pour l'accumulateur : vous saurez ainsi quelle valeur peut être utilisée dans les calculs



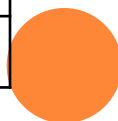
ASTUCES

- Dessiner une case à part pour retenir le numéro de l'instruction que l'ordinateur est rendu à traiter
- Dessiner une case à part pour retenir le code d'opération en cours
- Dessiner une case à part pour retenir la position en mémoire de l'opérande
- Ne pas oublier de tester le cas où on travaille avec le chiffre 0 ou une valeur négative

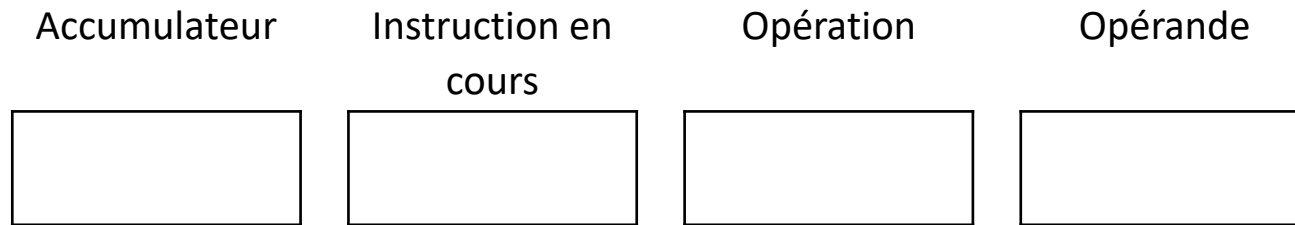


TABLEAU

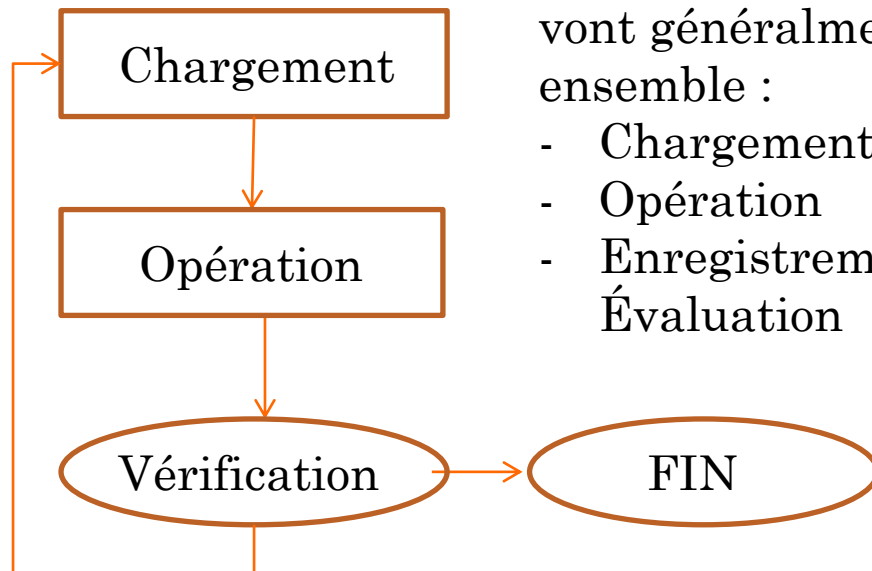
Position	Mot	Signification
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
90	? ?????	Variable A = nombre lu au clavier
91	+00000	Variable B = total (initialisée à 0 au départ)



ASTUCE



OU des bulles
d'instruction



Petite astuce : Voici 3 instructions qui vont généralement ensemble :

- Chargement
- Opération
- Enregistrement ou Évaluation

À chacun sa méthode



TRAVAUX PRATIQUES DE GROUPE

○ Exercice 1.1

Sur papier, écrivez un programme en LS qui lit des nombres tant qu'ils sont positifs. La lecture cesse dès qu'un nombre négatif est saisi. Le programme effectue ensuite leur somme puis affiche le résultat.

○ Exercice 1.2

Sur papier, écrivez un programme en LS qui lit 7 nombres à l'aide d'une boucle. Les nombres peuvent être positifs ou négatifs. Le programme affiche ensuite leur moyenne.



FICHE D'AUTOÉVALUATION

Question 1: Selon vous , qu'est ce qu'un Langage Machine ? Citez-en un que vous connaissez.

.

.

.

.

.

.

.

Question 2: Quelles differences faites vous entre un compilateur et un interpreteur ? Pouvez vous citez 2 langages gérées par un compilateur et 2 laganges gérées par un interpreteur ?

.

.

.

.

.

.

.

.

.

.

.



FICHE D'AUTOÉVALUATION

Question 3: Quelles sont les différentes composantes d'un Simpletron ?

.....

.....

.....

.....

.....

.....

.....

.....

Question 4: Combien de lignes d'instructions peuvent contenir un simpletron ? 00 01 02 ...

.....

.....

.....

.....

.....

.....

.....

.....



À FAIRE POUR LA PROCHAINE FOIS

○ Exercice 1.3

Sur papier, écrivez un programme en LS qui lit une série de nombres puis affiche le plus grand des nombres. Le premier nombre saisi indique combien de nombres seront saisis au total.

