

Pourquoi Ada ?

1) Introduction

Qu'est ce qui me pousse à exprimer ma préférence pour le langage informatique Ada, pour le moins de la partager, voire de la recommander ? Je pourrais simplement faire référence à de très bon plaidoyers (en anglais) existants sur Internet (<http://www.adaic.org/whyada/index.html> par exemple). Curieusement, je n'ai pas trouvé de réquisitoire contre (serait-ce de la mauvaise foi inconsciente ?) Bref, mon propos sera plus personnel. Mon adhésion à Ada est la conséquence d'une histoire propre qui commence avec les débuts de la micro-informatique dans les années 80 du siècle dernier. Bien que pas encore papi, la naissance de la micro-informatique a été concomitante avec mon passage à l'âge adulte avec ses espoirs et sa soif d'apprendre des sujets qui n'avait jamais existé auparavant.

2) Le Basic

Comme pour beaucoup à cette époque, mon premier apprentissage d'un langage informatique a été le Basic (1963). À l'époque très populaire, il est présent sur la plupart des tous premiers micro-ordinateurs l'Apple II bien sûr mais aussi le Sinclair ZX81, le Commodore VIC-20, le Texas TI99/4A, l'Atari 800XL entre autres pour ceux qui me sont passés entre les mains et même les calculatrices évoluées comme la Sharp PC-1500. Il comporte peu d'instructions, qui, interprétées, permettent une inter-activité immédiate avec l'utilisateur. Son initiation est très rapide pour quantité de personnes. Le magazine Hebdogiciel publia de 1983 à 1987 des dizaines de programmes en Basic envoyés par les lecteurs.

Néanmoins, il s'avérera très vite limité malgré l'apparition de plusieurs Basics "étendus". L'apprentissage de l'assembleur devint ainsi plus que nécessaire. S'il n'était pas aisé de programmer lisiblement en Basic (au sens de suffisamment lisible pour repérer rapidement - à la première ou au plus à la deuxième lecture - des erreurs de programmation parmi des dizaines de lignes de programmes), la vérité est que la structure des premiers Basics n'aidait pas. Les instructions de saut GOTO et les variables toutes globales déstructuraient le meilleur des programmes codé même avec la meilleur des volontés. Il en a été tout autre en assembleur.

3) L'assembleur

Avec l'assembleur, une méthodologie s'imposait : une erreur était généralement fatale pour le fonctionnement du micro-ordinateur, même pas droit au mythique "Syntax error" rudimentaire du Basic mais salvateur.

La pratique de l'assembleur était alors le summum de la reconnaissance. Il fallait alors non seulement connaître les instructions propres à chaque micro-processeur mais aussi le fonctionnement interne de son micro-ordinateur pour pouvoir utiliser les instructions correctement. Sa maîtrise permettait de s'affranchir des limitations du Basic dont la première était la lenteur. Quelle surprise, quelle joie, d'exécuter un programme assembleur pour s'apercevoir que sa rapidité d'exécution est quasi instantané remettant radicalement en cause la manière de le déverminer et donc notre façon de programmer.

La littérature de l'époque était somme toute très confidentielle. Quelques livres et magazines dont Hebdogiciel livraient de précieuses informations techniques mais pas de méthodologie.

Paradoxalement, une méthodologie de programmation avait tout son sens avec le langage assembleur, même, elle est bien plus facile d'application qu'en Basic car l'assembleur est à l'évidence plus souple. Cette méthodologie assez personnelle pour ma part s'apparentait à la programmation structurée : localisation des variables, bibliothèque de fonctions avec protocole d'entrée / sortie... Néanmoins, c'était laborieux d'écrire des lignes et des lignes d'instructions pour faire des choses somme toute simples et répétitives.

Puis vint Turbo Pascal...

Pascal Pignard, décembre 2010.