



Internet

Mille et un langages

Thomas Engel

Des langages de programmation, il y en a autant que de grains de sable au bord de la mer. Seuls quelques-uns sont vraiment connus. Tout le monde connaît en effet des langages tels que Java, C++, Basic ou Pascal; mais quels sont donc les autres? Quel a été le premier langage de programmation et comment l'a-t-on élaboré?

Histoire des langages de programmation

Il y a sur internet des milliers de pages consacrées au sujet des langages de programmation. On trouve une bonne synthèse de leur évolution à l'adresse <http://www.levenez.com/lang/lang.pdf>. Cette vue d'ensemble commence par le langage Fortran en 1954. Mais est-ce vraiment le premier? D'autres auteurs citent Lambda-calcul en tant que premier langage de programmation, et l'on entend souvent citer le nom d'Ada Lovelace en relation avec le premier «programme» en 1840. D'autres noms cités sont par exemple Plankalkül, A-0, Mark I Autocode, Flow-Matic ou Algol; ils se situent tous à la période entre 1945 et 1960. Mais quel est donc le véritable «premier» langage de programmation? Il est aussi difficile de donner la réponse qu'à la question de l'œuf et de la poule, surtout quand on n'a pas défini exactement ce que l'on entend par «programme».

Un autre moyen de trouver la réponse, c'est de considérer le matériel, c'est-à-dire l'ordinateur. Je ne vais pas remonter à Adam et Eve et aux premières aides mécaniques au calcul dans l'Antiquité. Commençons donc par les premiers calculateurs électromécaniques. Ceux-ci travaillaient avec des «programmes» câblés à demeure. Si l'on voulait modifier le programme, il fallait d'abord changer le câblage. On s'est vite rendu compte que ce n'était pas commode du tout et qu'il fallait trouver une autre solution. IBM a construit en 1934 la machine à cartes perforées IBM 601, dont le programme était perforé sur des cartes. Ce système binaire un-zéro est utilisé aujourd'hui encore.

Toutefois, le *bit* ne suffisait pas pour un langage de programmation. Il a fallu d'abord inventer le *byte* (ou octet), à savoir une combinaison de huit zéros et uns pour représenter une lettre ou un chiffre.

Dès lors, on a pu écrire des programmes en «code machine». Mais comme les machines «pensent» différemment de nous, la programmation en code machine est très lourde et quasiment impossible à maîtriser.

L'assembleur a été une première solution: il devenait possible de donner les instructions en tant qu'ordres simples. Mais l'assembleur est spécifique à chaque ordinateur et doit être réécrit pour chaque autre type d'ordinateur. Il est encore utilisé pour les machines à hautes performances, car c'est un langage très rapide et très direct.

C'est pourquoi, pour écrire des programmes indépendants du type d'ordinateur, on a développé ce qu'il est convenu de nommer des «langages de programmation de haut niveau», qui se rapprochent de la façon humaine de penser et qui peuvent s'exécuter sur différents ordinateurs pour autant que leurs normes soient respectées. En font partie, par exemple, les langages BASIC, PASCAL, C ou C++.

Vous pourrez découvrir à l'adresse <http://www.99-bottles-of-beer.net/> des centaines de langages de programmation accompagnés d'exemples, et en vous amusant par-dessus le marché!



Définitions

Bit: *binary digit* = chiffre binaire, généralement 0 ou 1.

Byte: octet ou mot, en informatique en général un groupe de huit bits, selon les codes. Exemples: ASCII: 1 caractère = 7 bits – Téléx: 1 caractère = 5 bits. Selon l'application, un mot peut comprendre un nombre différent de bits.

Code machine: format binaire de nombres contenant les instructions nécessaires à un processus. Comme le code machine dépend d'une machine donnée, il est difficile à maîtriser et à apprendre pour l'homme. Les compilateurs traduisent les langages évolués de programmation en code machine.

Algorithme: c'est la description de la méthode de résolution d'un problème, en divisant le programme en un nombre fini d'étapes. Le déroulement de traitement de chaque étape successive mène à la résolution du problème.

Compilateur: c'est un programme de traduction qui traduit un programme écrit dans un langage de programmation évolué en code machine compréhensible pour l'ordinateur.

Assembleur: langage de programmation spécifique à tel ou tel ordinateur comprenant le jeu d'instructions directes de cette machine. Ce langage direct permet d'accélérer les calculs des programmes à exécuter.

Debug: recherche des erreurs. Un *debugger* est un programme auxiliaire qui recherche les erreurs contenues dans un programme fraîchement écrit et en teste le bon fonctionnement. Le mot vient de l'anglais *bug* (vermine qui pouvait causer des erreurs dans les anciennes machines à tubes électroniques).

Langage de programmation de haut niveau: langage adapté à la manière de penser de l'homme.

Programme: collection d'instructions écrites dans un langage de programmation donné. Un programme sert à l'exécution d'instructions, d'étapes successives du traitement, d'opérations particulières ou d'impression de documents.

Code source: programme représenté en code ASCII d'un programme en langage de programmation de haut niveau. Il peut être traité à l'aide d'un éditeur de textes. Il ne peut fonctionner sous cette forme et doit d'abord être traduit en code machine par un compilateur.

Programmation orientée objet: méthode de programmation fréquemment utilisée aujourd'hui qui s'oriente selon des questions concrètes et non d'après une systématique informatique abstraite. Chaque objet peut ainsi être considéré comme un acteur indépendant qui résout un problème déterminé dans un système hiérarchiquement supérieur.

Je vous souhaite à tous une bonne année 2009!

A suivre...

