

Histoire du C

Le but de cette page n'est pas de raconter en détails l'histoire du C, mais plutôt de vous donner une idée de ce qu'est le langage que vous êtes en train d'apprendre, et ce de façon simple.

Cette page a été écrite à partir d'informations que j'ai récupéré un peu partout, et qu'il m'a fallu classer, trier, regrouper, ... Il est donc fort possible que, malgré tous mes efforts, cette page contienne des erreurs. Si tel est le cas, je vous prie de m'en excuser, et j'apprécierai fortement que vous m'écriviez pour me les signaler, afin que je puisse les corriger au plus vite. vous pouvez procéder de la même façon si vous pensez que j'ai oublié quoi que ce soit d'important. Merci.

La première apparition du langage de programmation nommé C s'est faite en 1972 dans les "Bell laboratories". Ce langage a été créé par Mr Dennis M. RITCHIE, afin de développer une version portable du système d'exploitation Unix (utilisé sur le DEC PDP-11). Depuis, ce langage, ainsi que son "frère" le C++ a connu une popularité croissante, qui explique pourquoi vous pouvez à présent l'utiliser pour programmer sur vos TI.

Nombre de principes de base du C sont issus du langage BCPL. Celui-ci à influencé le C par l'intermédiaire du langage B, inventé en 1970 par Ken Thompson pour le DEC PDP-7, qui fut le premier ordinateur à tourner sous système de type UNIX.

La définition classique du langage C a été donnée en 1978 par le duo composé de Brian W. KERNIGHAN et de Dennis M. RITCHIE, dans un livre intitulé The C Programming Language. (Ce livre est souvent abrégé sous la forme "K&R", et la définition du C qu'il donne est souvent appelée "Standard K&R-C")

Dans les années qui suivirent, le développement d'un grand nombre de compilateurs C a rendu nécessaire la création d'un nouveau standard, plus récent, et mieux défini. Ainsi, en 1983, le "American National Standards Institute" (que l'on peut abrégé en ANSI) a chargé une commission de donner une définition claire du C, mais qui devrait conserver l'esprit de ce langage. Le résultat fut la naissance du standard ANSI-C. Cette naissance fut suivie par la parution de la seconde édition du livre The C programming Language (en 1988), qui suit la définition du standard ANSI-C. Ce livre est d'ailleurs souvent considéré comme la Bible des programmeurs en C.

Comme tout langage de programmation, le C présente des avantages, mais aussi quelques inconvénients ; nous les résumerons rapidement ici, mais c'est à force de programmer que vous les sentirez, voire même en trouverez par vous-même, que je n'ai pas cité ici :

A: Les avantages :

- Universalité : Ce langage peut être adapté à de nombreuses machines, dans de nombreux buts. Pour preuve, il a bien été adapté pour les TI-89, TI-92plus, et V-200 !
- Modernité : Le C offre de nombreuses possibilités, comme des structures de répétition, de conditions, de déclarations, que l'on retrouve d'ailleurs dans la majorité des "grands" langages de notre temps.
- Rapidité : essayez donc de faire des programmes en TI-BASIC qui aillent à la même vitesse que leurs équivalents en C (Cela est dû au fait que le C est un langage compilé, à la différence du TI-BASIC, qui est un langage interprété.) !
- Extensibilité : Le C n'utilise pas seulement des fonctions standard, mais aussi des fonctions incluses dans les bibliothèques, qui permettent au langage d'être simple et concis, tout en autorisant les compilateurs à proposer un grand nombre de fonction, via les bibliothèques qu'ils fournissent au programmeur. Ainsi, ce langage est basé sur un petit nombre d'instructions, complétées par des fonctions (la version 2.3 des bibliothèques de TIGCC en regroupe environ 800), qui sont donc "peu nombreuses" à retenir, mais qui permettent de faire un grand nombre d'opérations, et ce en manipulant uniquement des opérations simples, ou que l'on arrive à considérer comme telles une fois qu'on a acquis un minimum d'expérience.

B: Les inconvénients :

Aucun langage de programmation ne peut être parfait. Le C n'échappe naturellement pas à cette règle. Le C, bien que généralement assez facile à programmer, est un langage qui peut comporter des "pièges".

En effet, il n'est pas rare de commencer un programme assez complexe, de le laisser momentanément (pour cause de vacances, par exemples), et de vouloir le finir deux semaines plus tard. Le risque est que, au bout de cette période, le programme soit devenu incompréhensible, pour cause d'enchaînement illogiques de fonctions.

Il vaut donc souvent mieux placer un grand nombre de commentaires dans ses programmes, quitte à perdre un peu de temps, plutôt que de vouloir aller vite, et, au final, en perdre encore plus.

Notons que le C propose l'instruction goto, redoutée par les puristes, dont l'usage est assez incorrect d'un point de vue algorithmique.