

Comment fonctionne l'interpréteur d'un langage de programmation ?

Jean Dubois

30 avril 2024

1 Introduction

Les ordinateurs sont des machines traitant des données et pouvant être programmées. Dès les premiers pas de l'informatique, et notamment avec la machine analytique imaginée en 1834 par Charles Babbage, dont le premier algorithme a été écrit par la première informaticienne de l'humanité : Ada Lovelace. Cette machine ne fut jamais construite. Les programmes écrits par Ada Lovelace étaient rédigés en langage mathématique, et non dans un langage de programmation tel qu'on en connaît aujourd'hui. Cependant, elle fut la première à comprendre la différence entre matériel et logiciel. [1, p. 454]

Les premiers ordinateurs électroniques, tel que l'ENIAC (Electronic Numerical Integrator and Computer), se programment en connectant physiquement les différents modules entre eux. [2] Après cela, les ordinateurs ont progressivement adopté un langage machine, où chaque instruction est représentée par un nombre, puis un langage assembleur, qui fait correspondre chaque instruction à un mot-clé mnémotechnique. [3]

Le premier vrai langage de programmation à proprement parler est Fortran, pour mathematical FORMula TRANslating system, créé en 1954 pour le calcul scientifique [1, p. 455]. Fortran était un langage compilé et haut-niveau, qui essayait au mieux de reproduire la notation mathématique. Jusqu'alors, un programme était une suite d'instructions pas-à-pas pour le processeur. En Fortran, l'idée était plutôt décrire le problème que l'on veut résoudre [4], et cette idée va être à l'origine de tous les autres langages de programmations jusqu'à nos jours.

Fortran est ensuite suivi par LISP (LISt Processor), qui est le premier langage interprété.

Nous allons nous intéresser dans cet exposé au fonctionnement des langages interprétés, bien que nous parlerons également des langages compilés. Nous expliciterons d'abord les points communs et les différences entre ces deux types de langage. Nous verrons ensuite dans le détail comment un interpréteur fonctionne. Enfin, nous nous intéresserons à comment implémenter un interpréteur en langage Python, et notamment en nous intéressant à l'interpréteur du langage Nougaro, écrit par mes soins.

2 Langages interprétés et langages compilés

Références

- [1] Gérard BERRY. *L'Hyperpuissance de l'informatique. Algorithmes, données, machines, réseaux*. Odile Jacob, 2017. ISBN : 9782738139535.
- [2] Paul A. FRIEBERGER et Michael R. SWAINE. *ENIAC*. 2008. URL : <https://www.britannica.com/technology/ENIAC>. (accédé le 2024/04/30).
- [3] William L. HOSCH. *Assembly Language*. 2009. URL : <https://www.britannica.com/technology/assembly-language>. (accédé le 2024/04/30).
- [4] *John Backus*. URL : <https://www.ibm.com/history/john-backus>. (accédé le 2024/04/30).

Licence

Ce document est placé sous licence GNU Free documentation license (FDL). Cela signifie que quiconque peut le copier et le redistribuer librement, avec ou sans modifications, dans un but commercial ou non, à condition de placer le document copié et/ou redistribué sous licence GNU FDL ou sous une licence compatible.

Plus précisément, ce document est placé sous les termes de la GNU FDL version 1.3 ou ultérieure, disponible en ligne à l'adresse www.gnu.org/licenses/fdl-1.3.html. Une copie du document de licence est normalement distribué avec ce document. Si ce document venait à être redistribué au format PDF, l'auteur apprécierait particulièrement la redistribution du code L^AT_EX en plus du PDF.