

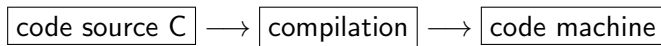
Langages compilés et interprétés

Prof d'info

Lycée Thiers

Un programme en C

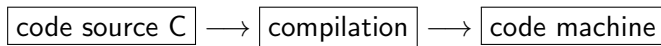
Nous avons vu qu'un programme écrit en C passe par plusieurs étapes :



- Le code machine est constitué d'instructions directement exécutables par le processeur.

Un programme en C

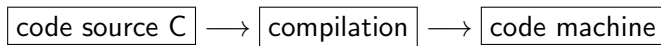
Nous avons vu qu'un programme écrit en C passe par plusieurs étapes :



- Le code machine est constitué d'instructions directement exécutables par le processeur.
- Il peut être enregistré dans un fichier exécutable.

Un programme en C

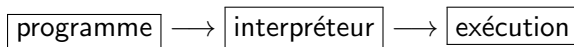
Nous avons vu qu'un programme écrit en C passe par plusieurs étapes :



- Le code machine est constitué d'instructions directement exécutables par le processeur.
- Il peut être enregistré dans un fichier exécutable.
- Une fois l'exécutable produit, il n'est plus nécessaire de disposer du code source ni du compilateur pour l'exécuter.

Un langage interprété

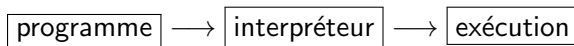
Une autre possibilité consiste à utiliser un programme appelé **interpréteur**.



- L'interpréteur lit le programme et détermine les opérations qu'il faut effectuer.

Un langage interprété

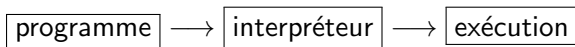
Une autre possibilité consiste à utiliser un programme appelé **interpréteur**.



- L'interpréteur lit le programme et détermine les opérations qu'il faut effectuer.
- C'est l'interpréteur qui est lui-même exécuté par le processeur.

Un langage interprété

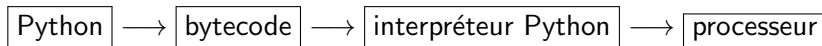
Une autre possibilité consiste à utiliser un programme appelé **interpréteur**.



- L'interpréteur lit le programme et détermine les opérations qu'il faut effectuer.
- C'est l'interpréteur qui est lui-même exécuté par le processeur.
- On ne produit donc pas nécessairement un fichier contenant directement le code machine du programme.

Exemple : Python

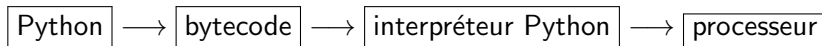
La situation réelle est souvent un peu plus compliquée.
Avec l'implémentation usuelle de Python (CPython) :



- Le code Python est d'abord traduit en instructions intermédiaires, appelées **bytecode**.

Exemple : Python

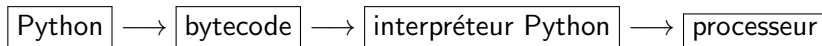
La situation réelle est souvent un peu plus compliquée.
Avec l'implémentation usuelle de Python (CPython) :



- Le code Python est d'abord traduit en instructions intermédiaires, appelées **bytecode**.
- Ces instructions ne sont pas directement des instructions du processeur.

Exemple : Python

La situation réelle est souvent un peu plus compliquée.
Avec l'implémentation usuelle de Python (CPython) :

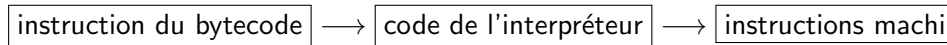


- Le code Python est d'abord traduit en instructions intermédiaires, appelées **bytecode**.
- Ces instructions ne sont pas directement des instructions du processeur.
- Elles sont exécutées par l'interpréteur Python.

Que fait réellement l'interpréteur ?

L'interpréteur est lui-même un programme compilé en langage machine.

Pour simplifier, lorsqu'il rencontre une instruction du bytecode, il exécute le code machine correspondant à cette instruction.



Le processeur n'exécute donc jamais directement du Python : il exécute l'interpréteur Python.

Compilé ou interprété ?

- On dit habituellement que le C est un langage **compilé** et que Python est un langage **interprété**.

Compilé ou interprété ?

- On dit habituellement que le C est un langage **compilé** et que Python est un langage **interprété**.
- Cette distinction est utile, mais elle est simplificatrice.

Compilé ou interprété ?

- On dit habituellement que le C est un langage **compilé** et que Python est un langage **interprété**.
- Cette distinction est utile, mais elle est simplificatrice.
- Un même langage peut avoir plusieurs modes d'exécution.

Compilé ou interprété ?

- On dit habituellement que le C est un langage **compilé** et que Python est un langage **interprété**.
- Cette distinction est utile, mais elle est simplificatrice.
- Un même langage peut avoir plusieurs modes d'exécution.
- Il existe également des situations intermédiaires : compilation vers un bytecode, machine virtuelle, compilation à la volée, etc.

Compilé ou interprété ?

- On dit habituellement que le C est un langage **compilé** et que Python est un langage **interprété**.
- Cette distinction est utile, mais elle est simplificatrice.
- Un même langage peut avoir plusieurs modes d'exécution.
- Il existe également des situations intermédiaires : compilation vers un bytecode, machine virtuelle, compilation à la volée, etc.

À retenir

La compilation traduit un programme avant son exécution ; l'interprétation fait intervenir un autre programme lors de son exécution.