

# Compilateurs : Introduction

Matthieu Amiguet

2009 – 2010



---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

- ① Motivations
- ② Définitions
- ③ Structure d'un compilateur
- ④ Autour du compilateur...

## Buts

3

- Comprendre la structure d'un compilateur
- Comprendre comment ses composants fonctionnent
- Faire connaissance des outils impliqués dans le développement de compilateurs
- Être capable de développer un compilateur simple.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## Motivations

4

- Curiosité
- Il faut bien que quelqu'un écrive des compilateurs
- Algorithmes à utilisation plus large
- Amélioration des pratiques de programmation.

- 1 Motivations
- 2 Définitions
- 3 Structure d'un compilateur
- 4 Autour du compilateur...

---

---

---

---

---

---

---

---

---

---

Compilateurs : Introduction  
Définitions

## Plusieurs langages... 7

- Le langage de départ est appelé *langage source*
- et celui d'arrivée le *langage objet* ou *langage cible*
- Il y a un troisième langage en jeu : le *langage d'implémentation* du compilateur
- Si le langage d'implémentation est le même que le langage source, on parle d'*autocompilation* (ang : *bootstrapping*).

---

---

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

Compilateurs : Introduction  
Définitions

## Définition 6

### Compilateur : définition générale

Programme qui prend en entrée le texte d'un programme dans un certain langage et donne en sortie le texte dans un autre langage...

- ... en préservant la signification !
- C'est un donc essentiellement un processus de *traduction*
- La plupart du temps, on sous-entend que
  - La source à traduire décrit un programme exécutable. Sa "signification" est donc à comprendre dans ce sens (*semantique opérationnelle*)
  - Le langage de départ est de plus "haut niveau" que le langage d'arrivée

---

---

---

---

---

---

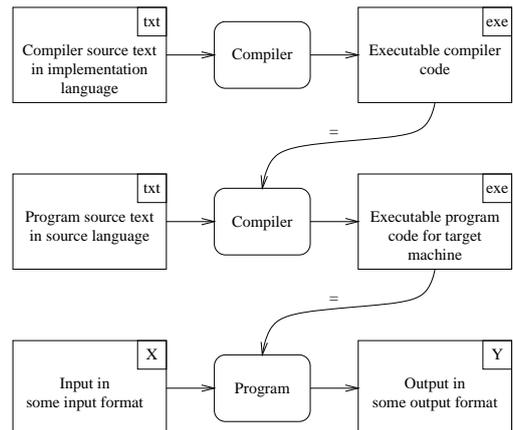
---

---

---

---

Compilateurs : Introduction  
Définitions



- 1 Motivations
- 2 Définitions
- 3 Structure d'un compilateur
- 4 Autour du compilateur...

---

---

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

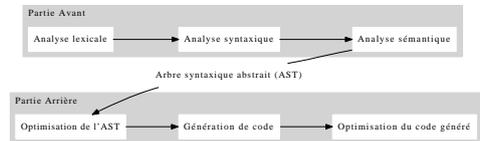
---

---

Compilateurs : Introduction  
Structure d'un compilateur

## Structure "idéale" d'un compilateur

10



- Dans un monde idéal, ces étapes seraient indépendantes
- Dans les faits, presque toujours un compromis (optimisation, efficacité, ...).

---

---

---

---

---

---

---

---

---

---

Compilateurs : Introduction  
Structure d'un compilateur

## Interprète

11

- Si la partie arrière exécute les actions sémantiques plutôt que de générer du langage objet, le programme est appelé *interprète*.
- Avantages
  - Souvent meilleure portabilité (langage de haut niveau)
  - Plus facile à programmer
  - Sécurité
- Limites
  - Vitesse
  - L'interprète lui-même doit être compilé !
- Compromis courant : la machine virtuelle à *bytecode*.

---

---

---

---

---

---

---

---

---

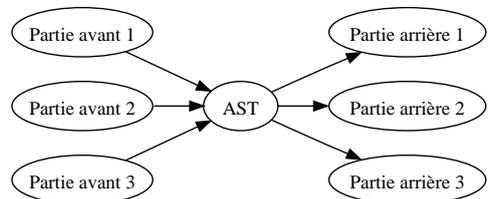
---

Compilateurs : Introduction  
Structure d'un compilateur

## Les polyglottes...

12

- Certains compilateurs sont capables de gérer plusieurs langage source et/ou objet



## Qu'est-ce qu'un *bon* compilateur ?

13

- Capital
- Important
- Dépend du contexte

---

---

---

---

---

---

---

---

- 1 Motivations
- 2 Définitions
- 3 Structure d'un compilateur
- 4 Autour du compilateur...

---

---

---

---

---

---

---

---

## Transportabilité et recyclage

14

- (Trans)portabilité : facilité de transport et de recompilation du code d'un type de machine à un autre
  - Comme tout programme informatique, un compilateur peut être plus ou moins (trans)portable
- Différent de la notion de *recyclage* : adaptation de la génération de code pour une autre machine
  - Nécessite la réécriture d'une partie arrière
  - Facilité par une implémentation propre...

---

---

---

---

---

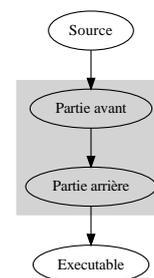
---

---

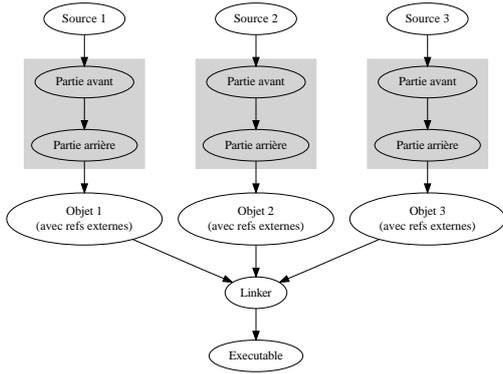
---

## Compilateur "tout seul"

16



# Éditeur de liens (Linker)



---

---

---

---

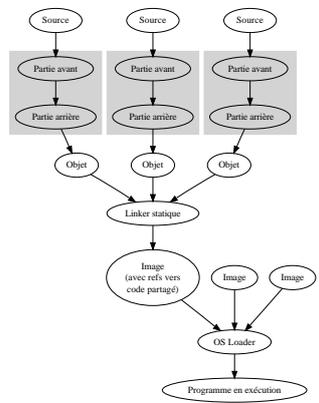
---

---

---

---

# Édition des liens dynamique



---

---

---

---

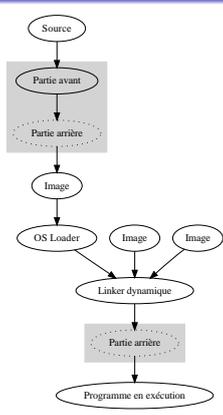
---

---

---

---

# Machine virtuelle



# Pour en savoir plus

- A. V. Aho, R. Sethi, J. D. Ullman, "Compilers : Principles, Techniques, and Tools", Addison-Wesley, 1986
- D. Grune, H.E. Bal, C.J.H Jacobs, K.G. Langendoen, "Compilateurs", Dunod, 2002
- Michael L. Scott, "Programming Language Pragmatics", Morgan Kaufman, 2000
- Lothar Schmitz, "What every computer scientist should know about syntax and syntax tools", disponible à l'adresse <http://inf2.w3.rz.unibw-muenchen.de/Tools/Syntax/english/theoryScreen.pdf>
- <http://www1.cs.columbia.edu/~sedwards/classes/2007/w4115-summer/compiler.ppt>