

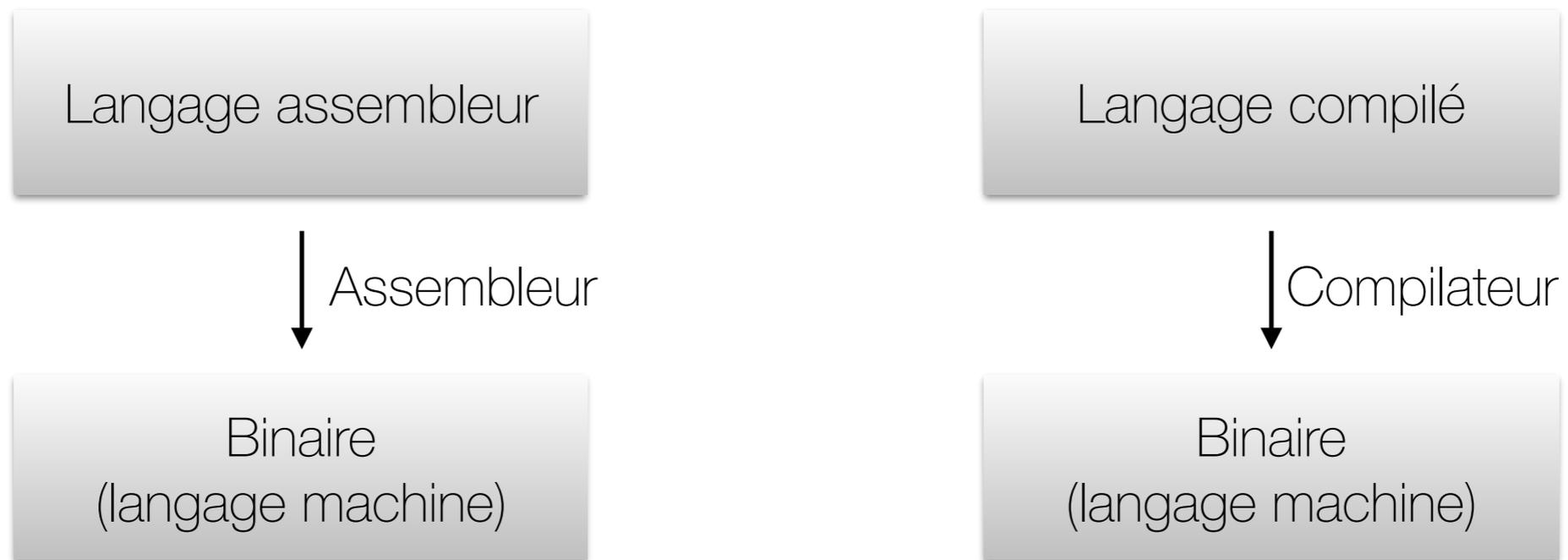
# Assembleur, Compilateur et Éditeur de Liens



Pierre de rosette

GIF-1001 Ordinateurs: Structure et Applications, Hiver 2015  
Jean-François Lalonde

# Assembleur et compilateur



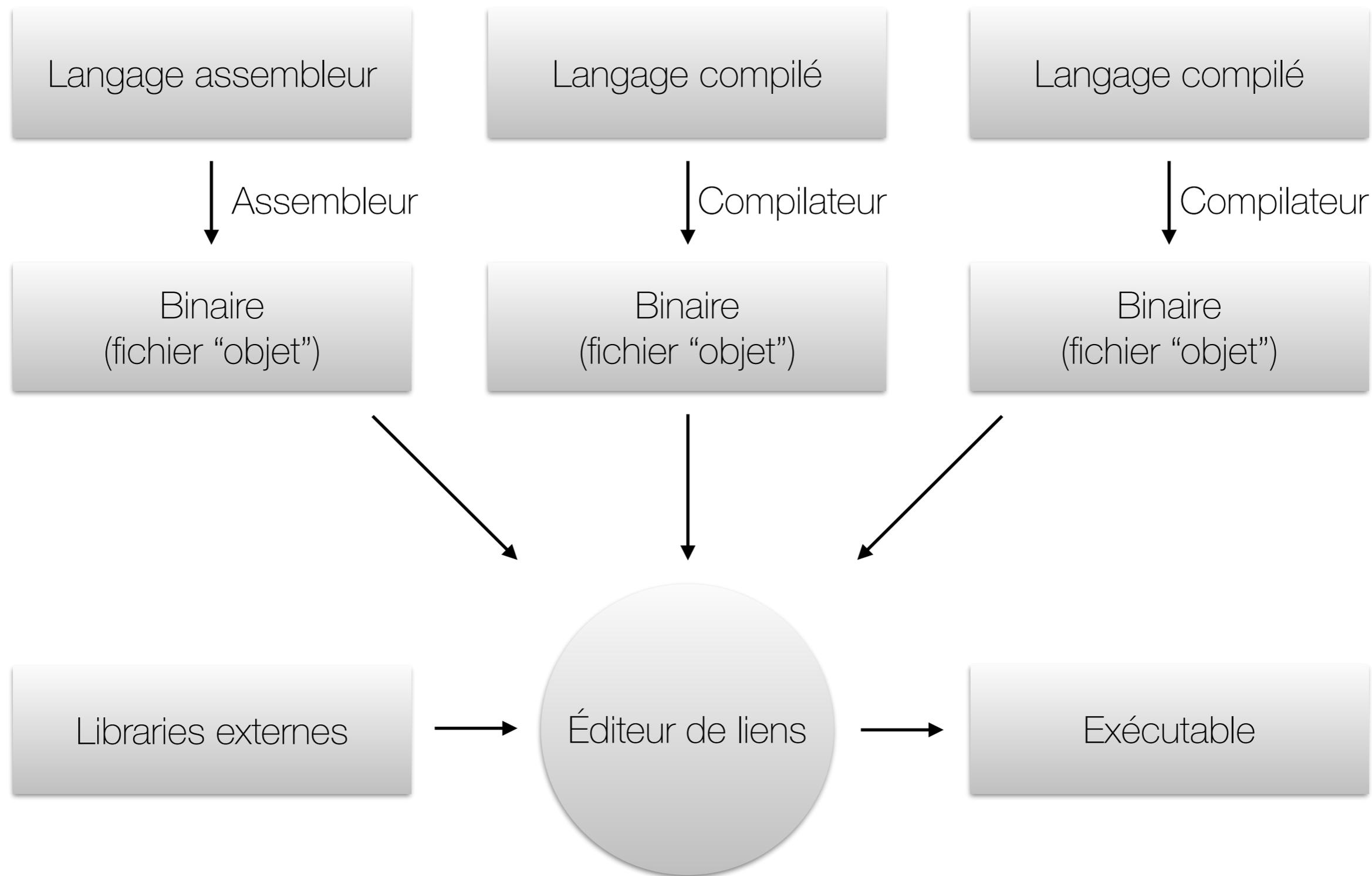
# Assembleur

- **Définition:** le langage assembleur permet d'écrire les instructions du microprocesseur en mnémoniques plutôt qu'en binaire (code machine). Le langage assembleur est traduit en code machine par un programme aussi nommé "assembleur".
- **But principal:** Faciliter l'écriture de programmes *pour un microprocesseur donné*.
- **Niveau d'abstraction et portabilité**
  - L'assembleur est très près du code machine. Une instruction d'assembleur correspond habituellement à une instruction machine.
  - Un programme en assembleur est indissociable du microprocesseur ou jeu d'instructions pour lequel il a été construit.
- **Exemples d'assembleur:** Il existe habituellement un assembleur pour chaque famille de microprocesseur.

# Compilateur et langage compilé

- **Définition:** un langage compilé est constitué de chaînes de caractère interprétées par un compilateur afin de générer de l'assembleur et, ultimement, du langage machine.
- **But principal:** Faciliter l'écriture de programmes et rendre les programmes disponibles *sur tous les microprocesseurs* pour lesquels il existe un compilateur.
- **Niveau d'abstraction et portabilité**
  - Les langages de haut niveau sont relativement indépendants du matériel. Le compilateur transforme habituellement une instruction de haut niveau en plusieurs instructions d'assembleur.
  - Un programme en langage de haut niveau peut être compilé pour être utilisé sur plusieurs microprocesseurs.
- **Exemples:** Fortran, Pascal, Basic, C, C++...

# Assembleur, compilateur et éditeur de liens



# Éditeur de liens (Linker)

- **Définition:** l'éditeur de liens rassemble toutes les fonctions et variables d'un programme afin de créer l'exécutable. Il agence les éléments du programme provenant de diverses sources (fichiers objets, bibliothèques dynamiques) pour que celui puisse être mis en mémoire par le système d'exploitation.
- **But principal:** Relier plusieurs segments de code machine afin de constituer un programme.
- **Buts secondaires**
  - Déterminer les emplacements en mémoire de chaque composante du programme.
  - Faire un lien entre toutes les fonctions et les appels de fonctions.
  - Faire un lien entre les variables globales de différents fichiers.
  - Faire un lien entre tous les fichiers du programme!
- **Niveau d'abstraction et portabilité**
  - Qu'on utilise un assembleur ou un compilateur, il existe toujours un éditeur de lien qui les accompagne...

# Langage interprété

- **Définition:** Un langage interprété n'est pas transformé en code machine. Il s'agit de fichiers textes qui sont traduits par un programme (habituellement nommé un engine) qui « agit » en fonction du contenu des fichiers.
- **But principal:** Permettre d'avoir des programmes (fichiers textes) totalement indépendants de la plateforme utilisée..
- **Niveau d'abstraction et portabilité**
  - Presque complètement indépendant du matériel.
  - Très haut niveau d'abstraction
- Exemples: HTML, Python, Matlab, ...

# Comparaisons

- Vitesse d'exécution
  - assembleur: très rapide!
  - compilé: très rapide, mais pas aussi rapide que de l'assembleur...
  - interprété: très lent...
- Grosseur du programme
  - assembleur: très petit
  - compilé: très gros
  - interprété: dépend de l'interpréteur
- Accéder au matériel
  - assembleur: naturel
  - compilé: faisable
  - interprété: ardu
- Temps de programmation
  - interprété: rapide
  - compilé: un peu plus long (généralement), mais néanmoins rapide
  - assembleur: éternel!