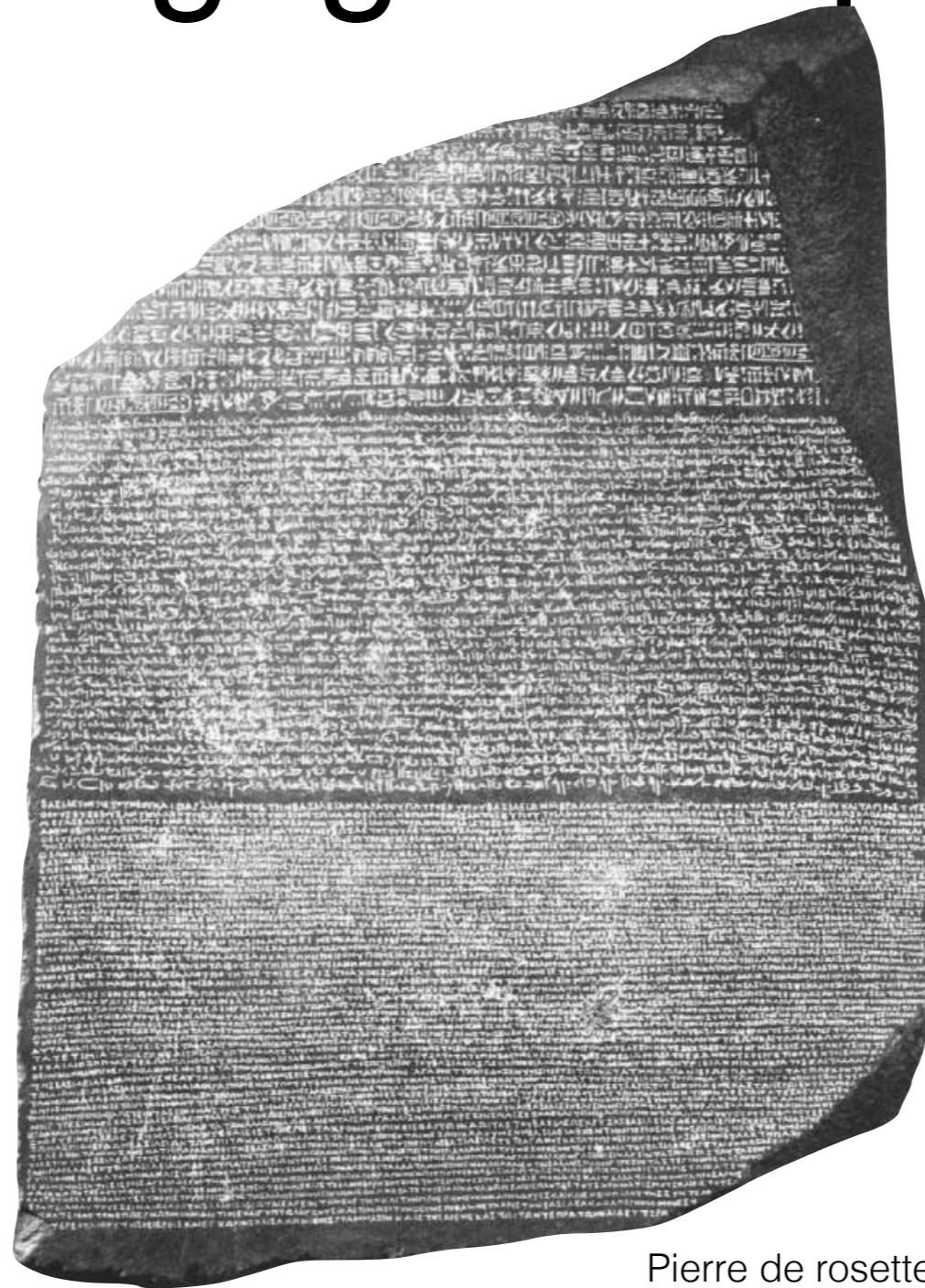


# Assembleur, compilateur et langage interprété



Pierre de rosette

GIF-1001 Ordinateurs: Structure et Applications, Hiver 2018  
Jean-François Lalonde

# Questions

- Où les instructions sont-elles stockées?
  - En mémoire!
- Comment représente-t-on une instruction dans un ordinateur?
  - En binaire, pardi!
- Comment faire pour stocker une instruction en binaire?
  - Nous aurons besoin d'une recette! Vous vous rappelez du PHIR™ #4?

# PHIR™ #4

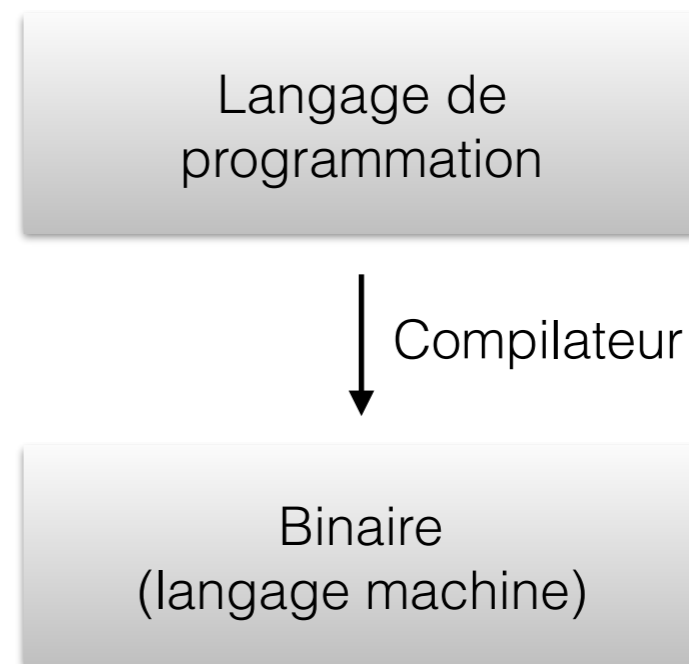
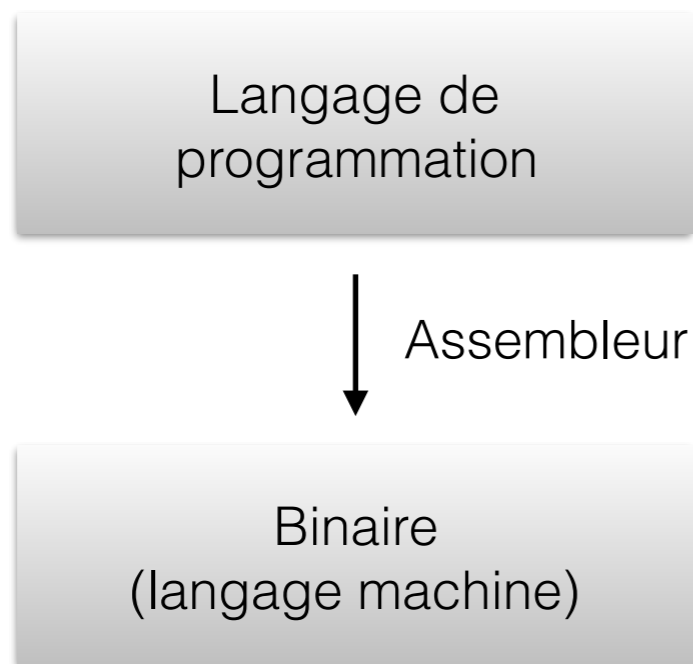
- A priori, nous ne pouvons pas savoir ce qu'une chaîne binaire signifie.
  - Ex: que veut dire 0xE3A00040 (sur 32 bits)?
  - La bonne réponse est: ça dépend!

entier non-signé	3818913856
entier signé	-476053440
IEEE754	-5.9...E21
assembleur ARM	MOV R0, #0x40

- Il nous *faut* donc savoir quel format (quelle «recette») utiliser pour bien interpréter les données!



# 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.

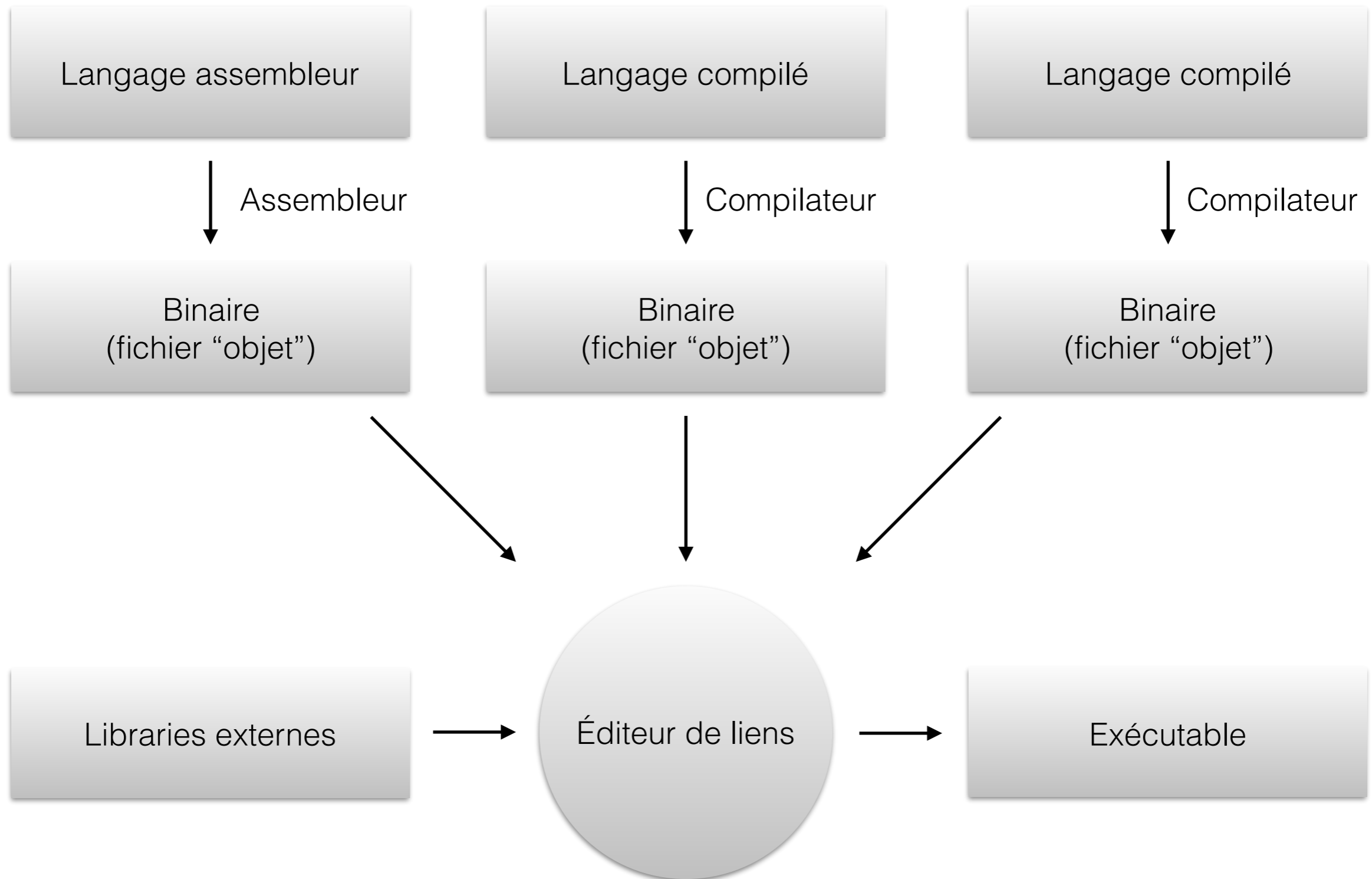
# Plus que de la «traduction» d'instructions

- En plus de traduire des mots/mnémoniques en binaire, l'assembleur interprète aussi le texte de plusieurs façons. Il permet:
  - d'associer des mots du programmeur à des adresses de mémoire.
  - au programmeur de déclarer des variables et il gère l'adresse de ces variables à travers les instructions du programme.
  - au programmeur d'identifier des fonctions ou des sections de codes avec des étiquettes (labels). Lorsque l'assembleur décode un appel de fonction ou un branchement (saut) dans le programme, il remplace les étiquettes par les adresses ou déplacements (offset) appropriées.
- L'assembleur supporte des directives qui lui disent comment placer le code en mémoire, comment gérer plusieurs fichiers, comment précompiler le code—modifier le code avant de le traduire en binaire—et plus. Les directives sont des mots réservés qui ne génèrent pas de code en binaire, mais qui dirigent la création du code exécuté.
- L'assembleur permet aussi d'insérer des commentaires dans le code!

# 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, finalement, 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 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

	Assembleur	Compilé	Interprété
Vitesse d'exécution	Le plus rapide!	Très rapide, mais pas autant que l'assembleur	Très lent...
Grosueur du programme	Très petit	Beaucoup plus gros	Dépend de l'interpréteur
Accéder au matériel	Naturel	Faisable	Plus difficile
Temps de programmation	Éternel...	Plus long qu'interprété, mais raisonnable	Très rapide