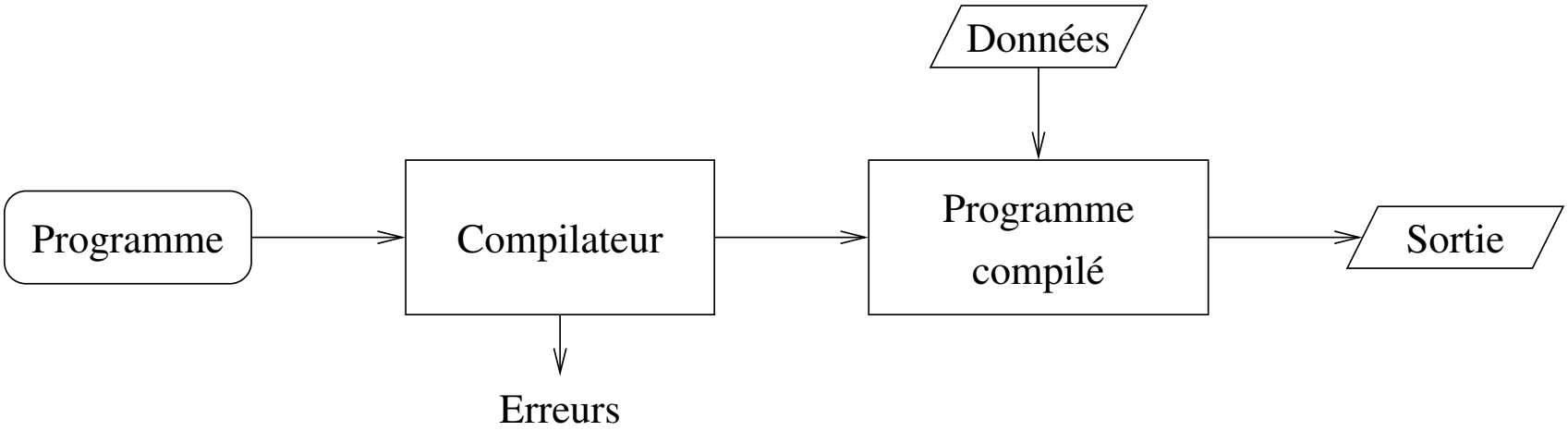
X31I090 - Langages et compilation
Interpréteurs et compilateurs

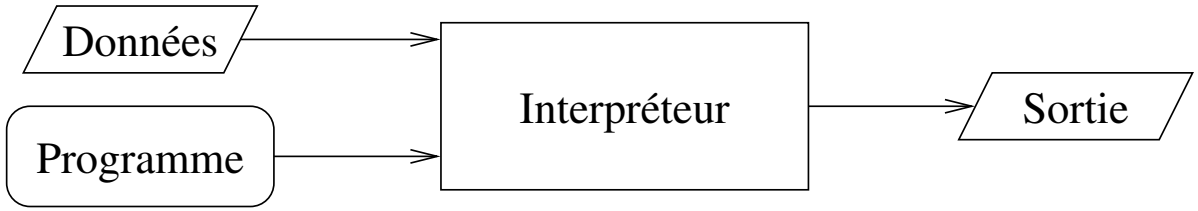
D. Béchet

Université de Nantes

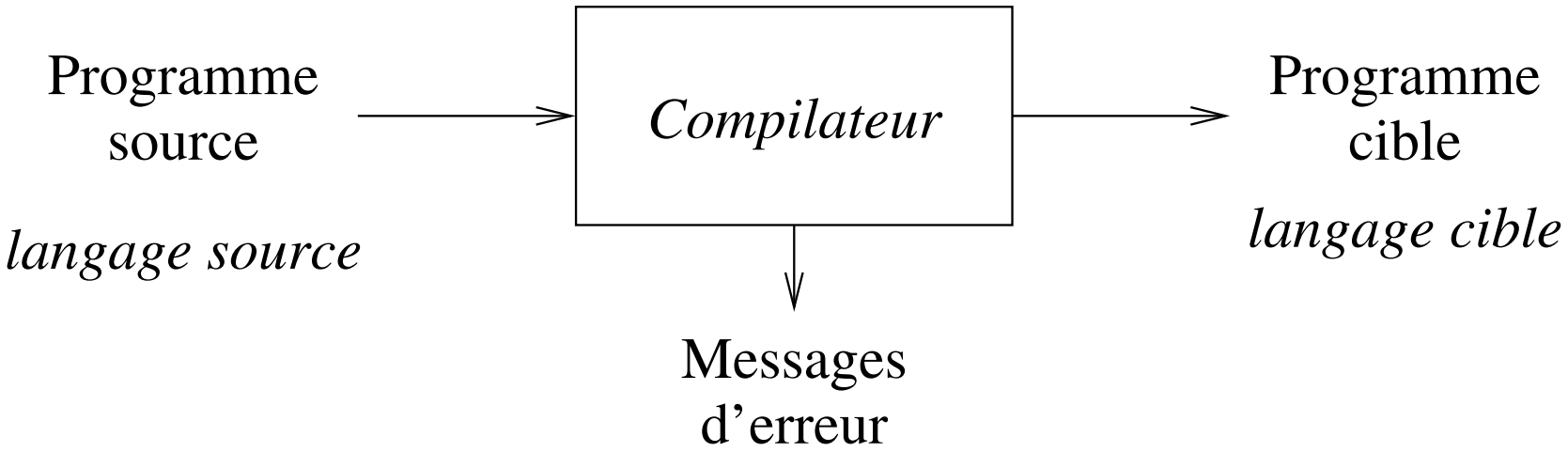
11 septembre 2018

Interpréteurs et compilateurs





Les compilateurs



Modèles de compilateurs

- Compilateur en une passe / en plusieurs passes
- Compilateur d'un langage vers :
  - le langage machine (spécifique à une architecture)
  - vers le code d'une machine virtuelle (plus général)
  - vers un autre langage (général)
- Compilateur optimisant
- ...

Compilateur généralement en deux parties :

Analyse du code source $\longrightarrow$ Génération du code cible

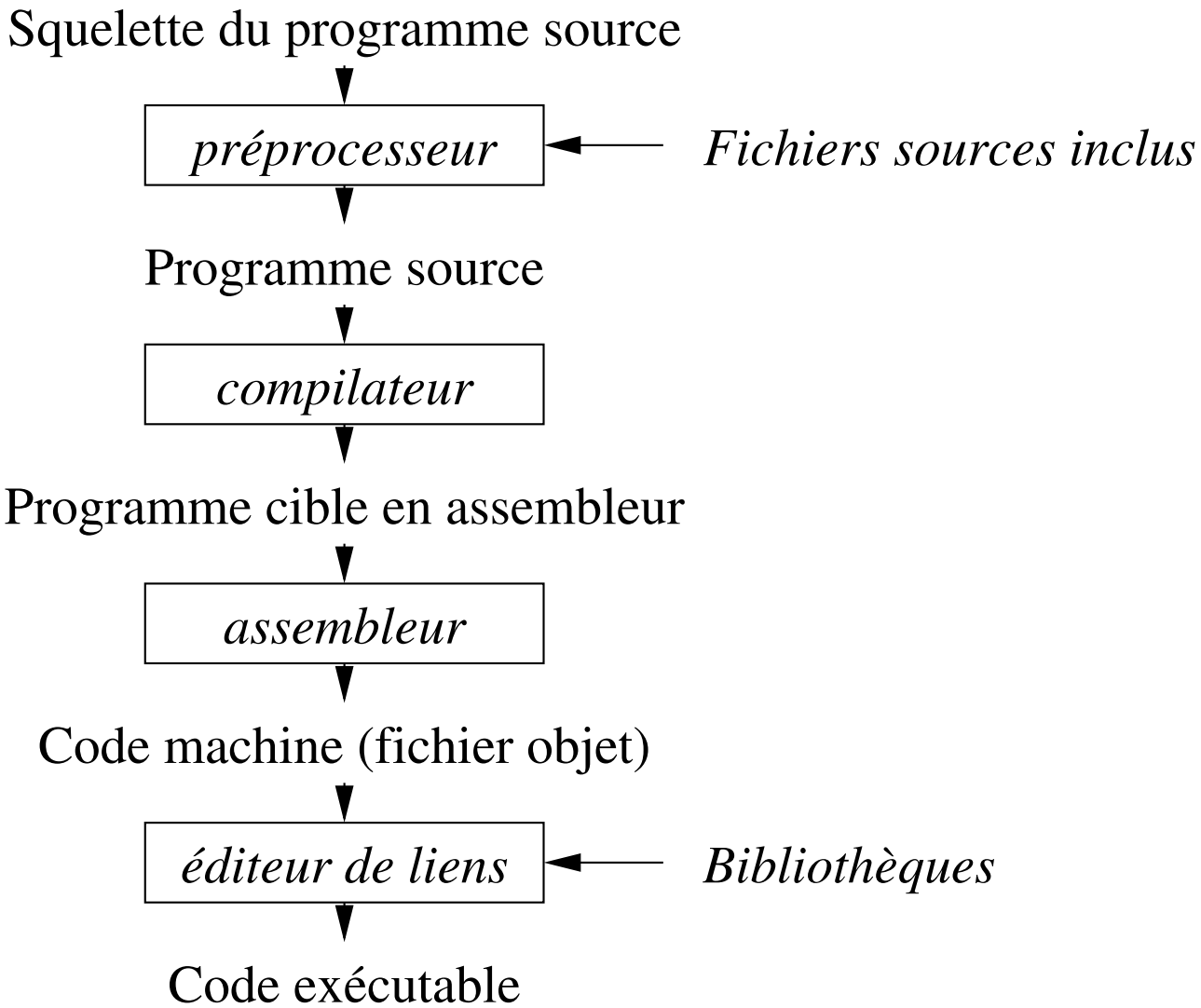
Interpréteur basé sur ce modèle :

Analyse du code $\longrightarrow$ Exécution des actions associées

Programme utilisant une forme d'analyse du code

- Éditeur de texte avec coloration du code
- Vérificateur syntaxique **lint** comme `pylint`, `php -l`
- Validateur (DTD d'un fichier XML)
- Analyseur statique de programmes effectuant une pseudo-exécution permettant de détecter certains défauts : pointeur nul, borne des tableaux...
- Interpréteur d'un langage de programmation
- "Compilateur" d'un langage source vers un langage cible
  - Compilateur générant un fichier exécutable (C, C++)
  - Compilateur générant un fichier exécutable par une machine virtuelle (C#, Java)
  - Composition d'un texte structuré (XML, RTF) vers un langage de description de pages (PDF)
  - Transformateur de document structuré (XML + feuille XSLT)
  - ...

"Compilateur" complet du type de gcc ou g++



Exemple de compilation

Code source exemple.c :

```
#include <stdio.h>
#define SALUER(x) printf("Bonjour %s !\n", x);
int main(int argc, char* argv[]) {
    char* nom = argv[1];
    SALUER(nom);
    return 0;
}
```

Compilation exemple.c $\longrightarrow$ exemple :

```
gcc exemple.c -o exemple
```

Exécution du programme exemple :

```
./exemple John
    ⟹ Bonjour John !
```

Exemple de compilation – prétraitement

Prétraitement exemple.c $\longrightarrow$ exemple.i :

```
gcc -E exemple.c >exemple.i
```

```
# 1 "exemple.c"
# 1 "<interne>"
# 1 "<command-line>"
...
# 212 "/usr/lib/gcc/x86_64-mageia-linux-gnu/4.9.2/include/stdc
typedef long unsigned int size_t;
...
# 2 "exemple.c" 2

int main(int argc, char* argv[]) {
    char* nom = argv[1];
    printf("Bonjour %s !\n", nom);;
    return 0;
}
```

Exemple de compilation – compilation du C

Compilation exemple.i $\longrightarrow$ exemple.s :

```
gcc -S exemple.i
```

```
.file "exemple.c"
.section .rodata
.LC0:
.string "Bonjour %s !\n"
.text
.globl main
.type main, @function
main:
...
movl $.LC0, %edi
movl $0, %eax
call printf
...
ret
...
```

Exemple de compilation – Assemblage

Assemblage exemple.s $\longrightarrow$ exemple.o :

```
gcc -c exemple.s
```

Voir la table symboles du fichier objet exemple.o :

```
objdump -t exemple.o
```

```
exemple.o:     format de fichier elf64-x86-64
SYMBOL TABLE:
0000000000000000 l    df *ABS*  0000000000000000 exemple.c
0000000000000000 l    d  .text  0000000000000000 .text
0000000000000000 l    d  .data  0000000000000000 .data
0000000000000000 l    d  .bss   0000000000000000 .bss
0000000000000000 l    d  .rodata        0000000000000000 .rodata
0000000000000000 l    d  .note.GNU-stack        0000000000000000 .note.GNU-stack
0000000000000000 l    d  .eh_frame      0000000000000000 .eh_frame
0000000000000000 l    d  .comment       0000000000000000 .comment
0000000000000000 g     F .text  0000000000000038 main
0000000000000000         *UND*  0000000000000000 printf
```

Exemple de compilation – Assemblage

Voir le langage machine (désassemblé) du fichier objet exemple.o :

```
objdump -dr exemple.o
```

```
Déassemblage de la section .text:
0000000000000000 <main>:
   0:   55                      push   %rbp
   1:   48 89 e5                mov    %rsp,%rbp
   4:   48 83 ec 20             sub    $0x20,%rsp
   8:   89 7d ec                mov    %edi,-0x14(%rbp)
   b:   48 89 75 e0             mov    %rsi,-0x20(%rbp)
   f:   48 8b 45 e0             mov    -0x20(%rbp),%rax
  13:   48 8b 40 08             mov    0x8(%rax),%rax
  17:   48 89 45 f8             mov    %rax,-0x8(%rbp)
  1b:   48 8b 45 f8             mov    -0x8(%rbp),%rax
  1f:   48 89 c6                mov    %rax,%rsi
  22:   bf 00 00 00 00          mov    $0x0,%edi
                        23: R_X86_64_32 .rodata
  27:   b8 00 00 00 00          mov    $0x0,%eax
  2c:   e8 00 00 00 00          callq  31 <main+0x31>
                        2d: R_X86_64_PC32 printf-0x4
  31:   b8 00 00 00 00          mov    $0x0,%eax
  36:   c9                      leaveq
  37:   c3                      retq
```

Exemple de compilation – Edition de liens

Edition de lien de exemple.o $\longrightarrow$ exemple :

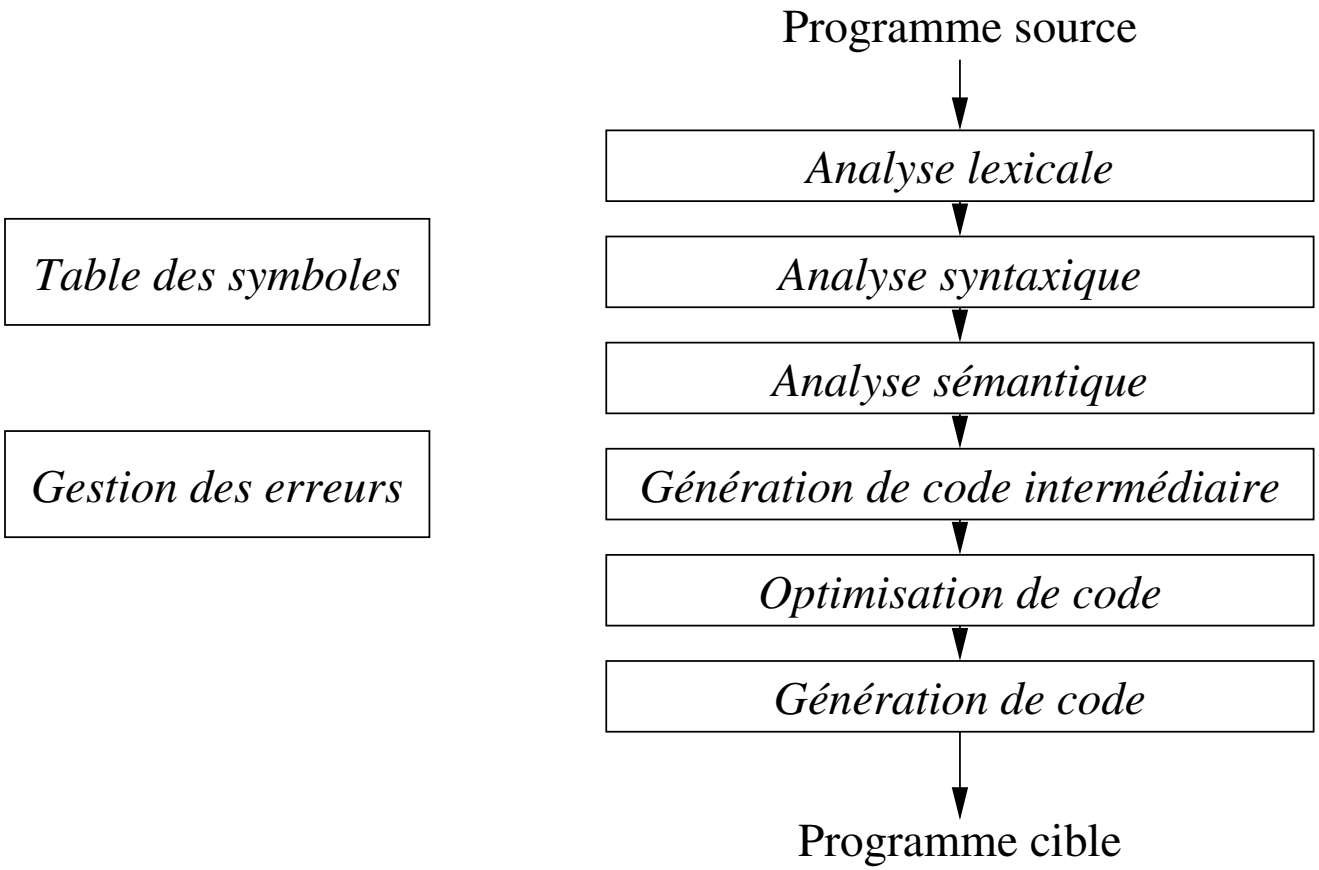
```
gcc exemple.o -o exemple
```

Voir la table symboles du fichier exécutable exemple :

```
objdump -t exemple
```

```
exemple:     format de fichier elf64-x86-64
SYMBOL TABLE:
0000000000000000 l    df *ABS*  0000000000000000              exemple.c
0000000000000000 l    df *ABS*  0000000000000000              crtstuff.c
...
0000000000601030  w      .data  0000000000000000              data_start
0000000000601040 g       .data  0000000000000000              _edata
00000000004006d4 g     F .fini  0000000000000000              _fini
0000000000000000       F *UND*  0000000000000000              printf@@GLIBC_2.2.5
...
0000000000400660 g     F .text  0000000000000065              __libc_csu_init
0000000000601048 g       .bss   0000000000000000              _end
0000000000400520 g     F .text  0000000000000000              _start
0000000000601040 g       .bss   0000000000000000              __bss_start
0000000000400628 g     F .text  0000000000000038              main
...
```

Phases d'un compilateur

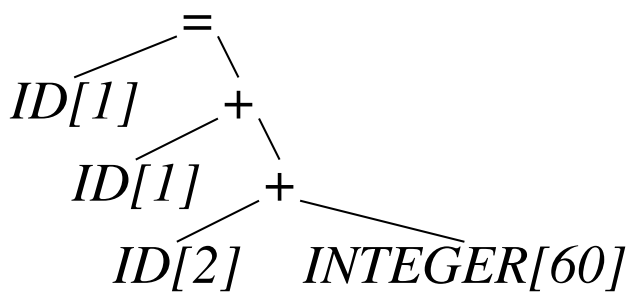


Phases d'un compilateur – exemple

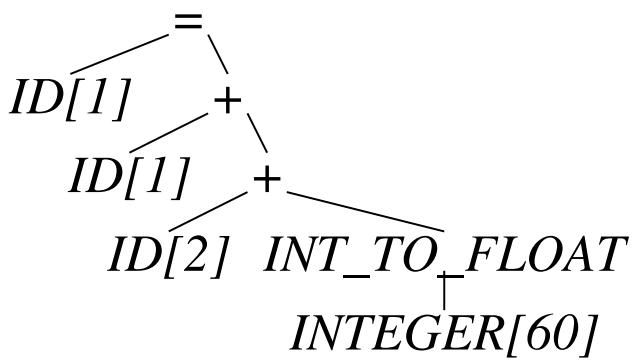
Programme source : `x = x+v*60;`

Analyse lexicale : *ID[1] = ID[1] + ID[2] * INTEGER[60] ;*

Analyse syntaxique :



Analyse sémantique :



Phases d'un compilateur – exemple

Code intérmédiare :

```
temp1 := INT_TO_FLOAT(60)
temp2 := ID[2] * temp1
temp3 := ID[1] + temp2
ID[1] := temp3
```

Code optimisé :

```
temp2 := ID[2] * 60.0
ID[1] := ID[1] + temp2
```

Code généré :

```
movf [x], R2
mulf #60.0, R2
movf [v], R1
addf R2, R1
movf R1, [v]
```

Phases d'un compilateur – exemple

Table des symboles :

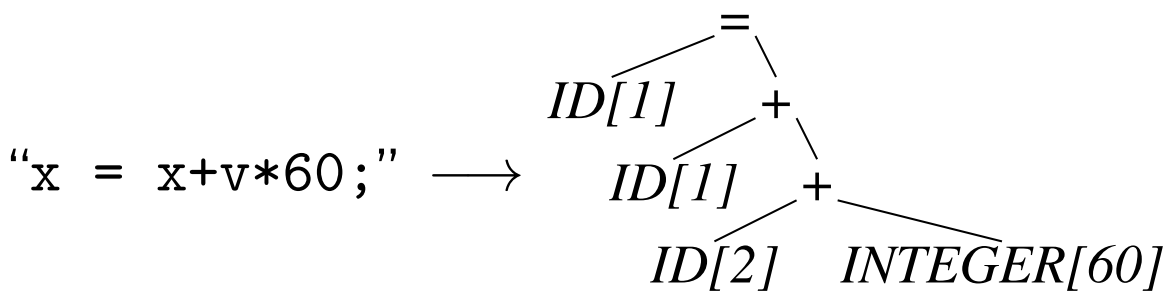
Symbole	Genre	Type	...
x	variable globale	float	
v	variable locale	float	...
main	fonction	int	

Erreurs :

- Erreur de syntaxe
- Variable ou fonction inconnue (non déclarée)
- Type incorrect : opération, paramètre de fonction, etc
- Avertissement : variable non initialisée, pas de `return`, etc

Suite du cours

Transformation d'une suite de caractères en une structure arborescente :



- Modélisation de la syntaxe des langages sources par des grammaires
- Analyseurs génériques
  - Analyse lexicale
  - Analyse syntaxique
  - Génération d'analyseurs à partir de grammaires