Programmation C/C++

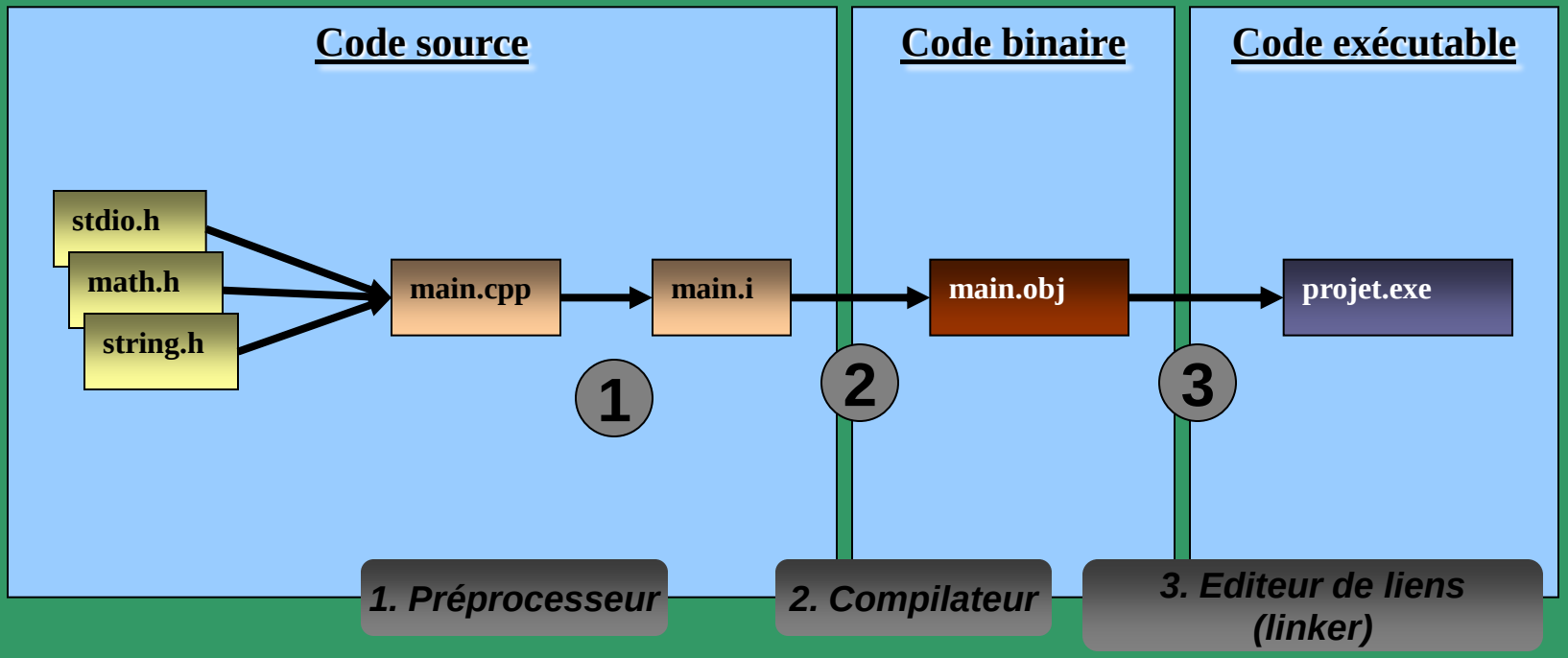
Compilation d'un projet mono-source

Nicolas Gazères

Dassault Systèmes
DS Research, Life Sciences
ngs@3ds.com

Compilation d'un projet mono-source

Schéma général



- A chaque étape correspondent des jeux d'erreurs très caractéristiques.

Le préprocessing d'un fichier-source

Constantes symboliques, macros, commentaires

- **1/ Constantes symboliques**
  - Lors du préprocessing, toute occurrence du symbole est substituée par sa valeur litérale.

```
#define   MAXSIZE    100
#define   PI     3.14159
```

- **2/ Macros**
  - Attention: c'est le premier espace qui détermine la fin de la partie gauche de la macro (algorithme glouton)

```
#define MAX(x,y)        (x)<(y)?(y):(x)
#define SQUARE(x)       ((x)*(x))
#define ISSPACE(x)      ((x)==' ')
```

- **3/ Commentaires**
  - Ce qui suit **//…** ou ce qui est entre **/*…*/**
  - Lors du préprocessing, les commentaires sont supprimés.

Exemple de Préprocessing

mon_source.c

```
#define MON_FORMATTAGE   "%d"
#define NB_MAX                  10
#define CARRE(x)                ((x)*(x))

int main() {
   // Cette boucle affiche la somme des carres
   // de 1 à NB_MAX

  unsigned int somme = 0 ;
  for (int i=1; i<=NB_MAX; i++ )
      somme += CARRE( i ) ;

  printf( MON_FORMATTAGE, somme );
  return 0;
}
```

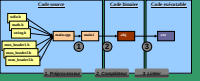
mon_source.i

```
#define MON_FORMATTAGE   "%d"
#define NB_MAX                  10
#define CARRE(x)                ((x)*(x))

int main() {
   // Cette boucle affiche la somme des carres
   // de 1 à NB_MAX

    unsigned int somme = 0 ;
    for (int i=1; i<=10; i++ )
        somme += ((i)*(i)) ;

    printf( "%d", somme );
    return 0;
 }
```



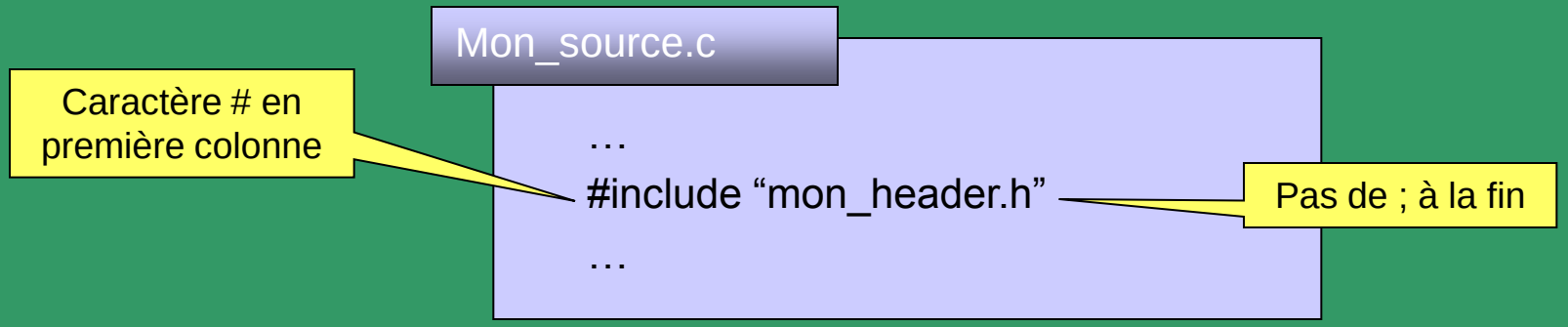
Qu'est-ce qu'un *Header* ?

- Définition
  - Un header est un *fichier* d'extension *.h* (en général).
  - Il contient des *déclarations de symboles:*
    - En C: noms des variables, de fonctions, …
    - En C++: même chose, et aussi des noms de classes, de templates…
- Le header est un contrat
  - destiné à tous les fichiers-sources qui utilisent les symboles déclarés dans ce header (les "clients").
  - qui garantit que tous les fichiers-sources "clients" voient la même déclaration de chaque symbole.
    - Si la déclaration *centralisée* dans le header est modifiée, la nouvelle version est visible dans tous les clients qui incluent le header
    - La modification sera prise en compte à la prochaine compilation.
- Exemples
  - Les headers standards (livrés avec le compilateur)
    - *stdio.h*, *stdlib.h*, *math.h, limits.h*…
  - Les headers d'un projet particulier
    - Un projet définit quasiment tout le temps ses propres headers (voir suite du cours).
  - Les headers d'un package externe
    - Si un projet particulier s'appuie sur un logiciel externe…

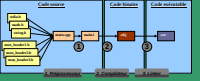
Le préprocessing d'un Fichier Source (2)

La directive d'inclusion de headers (#include)

- Syntaxe

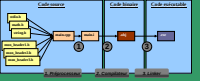


  - Le passage du préprocesseur sur le fichier-source a pour effet de remplacer la ligne d'inclusion par le contenu intégral du fichier *header.h*.
  - Il ne faut pas mettre les chemins complets des fichiers dans des directives d'inclusion
    - Sinon, le projet n'est plus partageable ni distribuable.
  - On indique au compilateur comment localiser les headers:
    - *Additional Include Directories (cf. prochain cours)*



Unité de Traduction

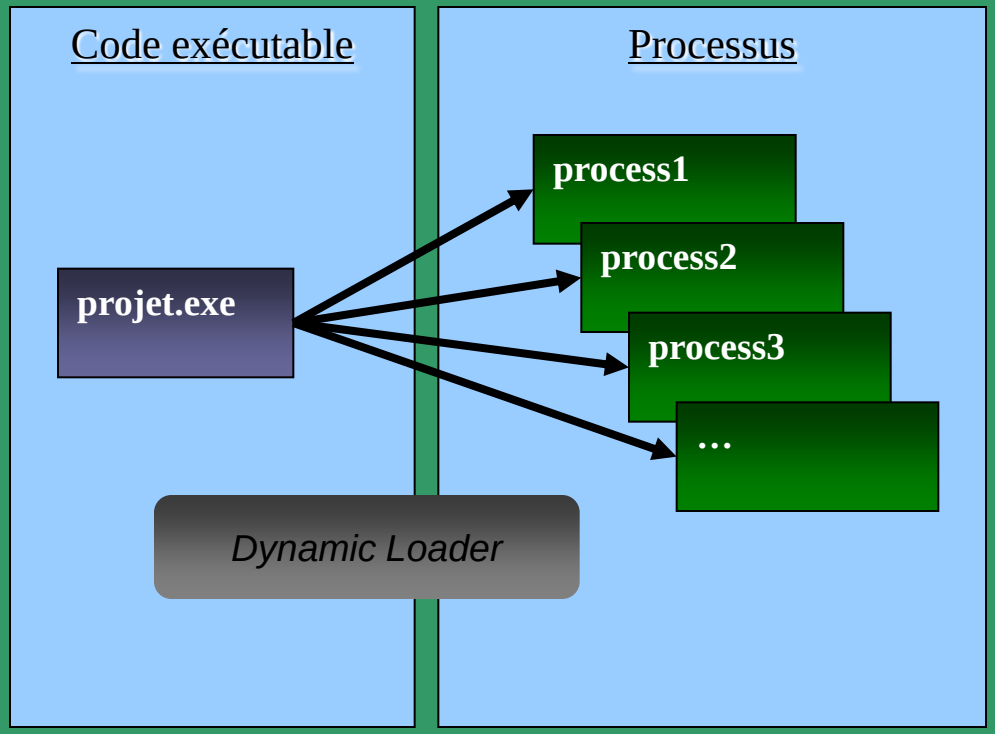
- Définition
  - C'est un fichier d'extension *.i* (en général)
  - C'est le fichier texte obtenu après l'action du préprocesseur sur le fichier source.
    - les macros ont été substituées.
    - Les commentaires ont été supprimés.
    - les inclusions de fichiers ont été réalisées.
    - le code n'est plus que du "pur" C (ou du pur C++).
- L'unité de traduction est de durée de vie très courte.
  - Elle est supprimée dès que le fichier objet résultant a été généré…
    - …sauf si on paramètre Visual Studio pour ne pas l'effacer.
- L'unité de traduction est un fichier prêt à être passé au compilateur.
  - Il est encore compréhensible par l'homme.



Fichier Objet

- Définition
  - C'est un fichier d'extension *.o ou .obj* (en général).
  - Un fichier objet est le fichier obtenu après le passage du compilateur sur une unité de traduction.
    - monSource.i → monSource.obj
- Il est écrit dans un format *binaire*.
  - non-compréhensible par l'humain.
  - Il est architecture-dépendant.
  - Un fichier objet *Windows* n'est pas compréhensible par un *Mac…*
- Il n'est pas exécutable (pas encore).
  - … sauf si le projet est composé d'un seul fichier source.

Lancement de l'exécutable



- L'exécutable peut être lancé plusieurs fois.
- Chaque process dispose de son propre espace mémoire.
- Chaque process démarre par le *main()*
  - *Attention*: uniquement quand le programme est écrit en C.
    - Cela ne sera plus forcément vrai en C++ (cas des objets *static* avec constructeur).

Compilation d'un projet mono-source

Récapitulatif

	Header	Source	Unité de Traduction	Objet	Exécutable
Extension du fichier	**.h**, .H, .hxx, .hh, …	**.cpp**, .c, .cxx, .C, .cc	.i	.o, .obj	.exe (ou aucune, sur Unix)
Présence de macros ?	Oui		Non	n/a	
Lisible par l'homme ?	Oui			Non	
Inclusion de Headers ?	Oui		Non	n/a	
Architecture-dépendant ?	Non			Oui (*)	
Exécutable ?	Non				Oui
Produit par	Un Editeur de texte		Le Préprocesseur	Le Compilateur	Le Link-Editor (linker)

(*) Conséquence pratique: un fichier objet généré sur un PC est inutilisable sur un Mac ou un Unix.

Fin