

Compilation



La fabrication d'un programme nécessite la transformation d'un **code source** écrit en langage C/C++ en **code objet** (ou **code machine**). C'est le rôle du **compilateur** (gcc/g++ par exemple).

La fabrication d'un (binaire) **exécutable** est réalisée en plusieurs étapes : préprocesseur (ou pré-compilation), assemblage, compilation et édition des liens.

Compilation (préprocesseur, assemblage et compilation) :

```
g++ -c -o main.o main.cpp
```

Édition des liens :

```
g++ -o main.out main.o
```

Linux ↔ Windows

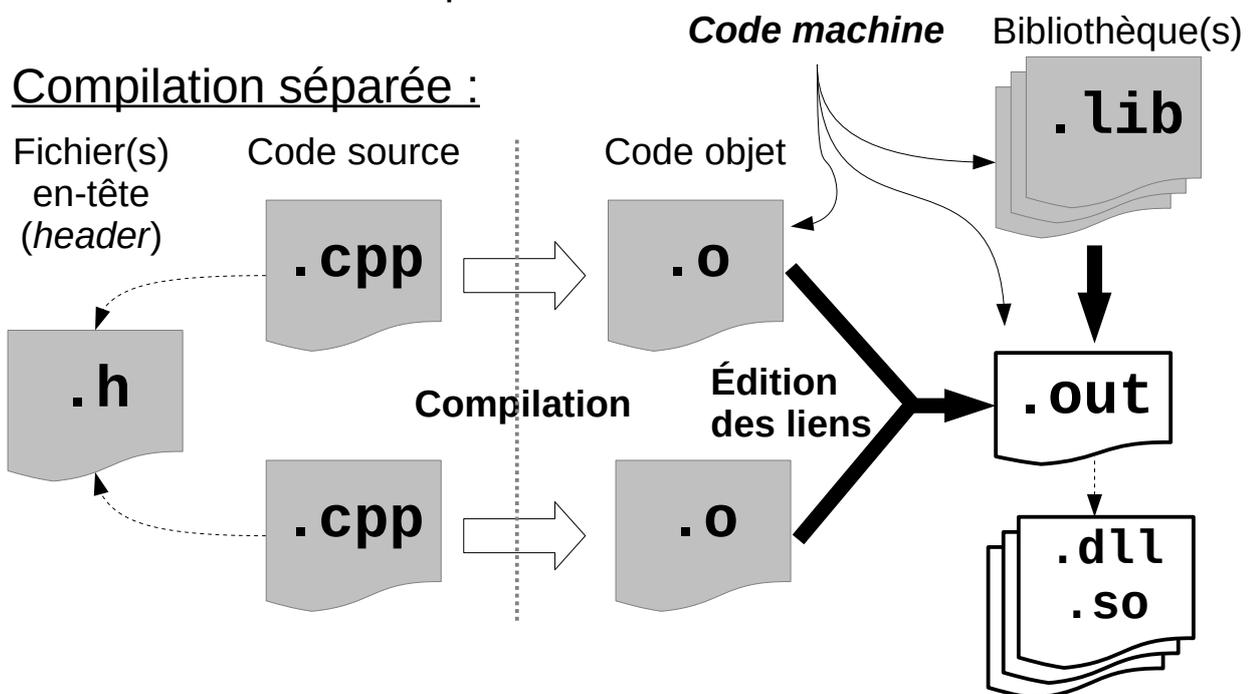
.o ↔ .obj

.out ↔ .exe

.so ↔ .dll

La **programmation modulaire** (découpage en plusieurs fichiers sources) entraînera la **compilation séparée**. La compilation séparée utilise la plupart du temps un outil (comme **make**) qui assure la fabrication automatique.

Compilation séparée :



Quand le programme compilé peut être exécuté sur un ordinateur dont le processeur (ou le système d'exploitation) est différent de celui du compilateur, on parle de **compilation croisée** (cross-compilation). C'est souvent le cas pour les **systèmes embarqués**.