

Les étapes de fabrication d'un exécutable peuvent être automatisées en utilisant l'outil **make** et en écrivant les **règles de fabrication** dans un fichier **Makefile**.

make est un outil indispensable aux développeurs car : il assure la compilation séparée automatisée, il permet de ne recompiler que le code modifié, il utilise un fichier distinct contenant les règles de fabrication (mémorisation des règles) et il permet d'assurer les tâches de nettoyage, d'installation, d'archivage, etc ...

Une **règle** est une suite d'**instructions** qui seront exécutées pour construire une **cible** si la cible n'existe pas ou si des dépendances sont plus récentes.

**cible: dépendance(s)
<TAB>commande(s)**

Une règle

```
CXX = g++

main: main.o calcul.o
    $(CXX) -o $@ $^

main.o: main.cpp calcul.h
    $(CXX) -c $<

calcul.o: calcul.cpp calcul.h
    $(CXX) -c $<

clean:
    rm -f main *.o
```

Makefile de base

Les variables automatiques :

\$@ : la cible
\$? : dépendances plus récentes que la cible
\$< : la première dépendance
\$^ : toutes les dépendances

Fabrication :

\$ make
\$ make main
\$ make clean