

# Guide *gcc* Pour l'Impatient(e)

“gcc - GNU project C and C++ compiler”, /usr/bin/man 2004.  
“Le Guide *gcc* Pour l'Impatient(e) est un manuel modérément restrictif de la compilo-synthèse.”,  
nj 2004.

Point de départ : un fichier source *Fsource* au format texte contenant un programme bien écrit.  
Objectif : compiler ce programme pour l'exécuter plus tard. Il suffit d'avoir en tête les principales options de la commande *gcc*.

Les options doivent être énoncées indépendamment les unes des autres. Les options **-dr** et **-d** **-r** ne sont pas équivalentes.  
L'ordre d'énumération des options est sans importance.  
L'option **--help** permet d'afficher la description des options de *gcc*.

## 1 Langages compilables avec *gcc*

L'option **-x langage** permet de spécifier le langage du programme utilisé dans *Fsource*. Les différentes occurrences possibles de **-x langage** sont :

- **-x c** pour un programme en *C*,
- **-x c++** pour un programme en *C++*,
- **-x objective-c** pour un programme en *Objectif-C*,
- **-x assembler** pour un programme en *assembleur*,
- **-x ada** pour un programme en *ADA*,
- **-x f77** pour un programme écrit en *fortran77*,
- **-x java** pour un programme en *Java*.

## 2 Détail des étapes de compilations

L'exécution de la commande *gcc* regroupe les phases de précompilation, compilation, assemblage et édition de liens. Ces phases sont respectivement réalisées par les commandes *cpp*, *cc1*, *as* et *ld* (pour un programme *C* sur Linux 2.4.22-10).

Il est possible de contrôler l'exécution de ces opérations à l'aide de l'extension du fichier **Fsource** :

- **file.c** est un programme *C* à précompiler,
- **file.i** est un programme *C* ne nécessitant pas de précompilation,
- **file.h** est un fichier d'entête *C* ne nécessitant pas de compilation et d'édition de liens,
- **file.cc file.cp file.cxx file.cpp file.c++ file.C** est un programme *C++* à précompiler,
- **file.ii** est un programme *C++* ne nécessitant pas de précompilation,
- **file.S** est un programme *assembleur* ne nécessitant pas de précompilation,
- **file.s** est un programme *assembleur* ne nécessitant pas de précompilation.

Il est également possible de contrôler ces opérations à l'aide d'options :

- **-E** pour n'appeler que le préprocesseur,
- **-S** pour compiler sans assemblage et sans édition de liens,
- **-c** pour supprimer la phase d'édition de liens,
- **-v** pour un mode verbeux affichant le détail des commandes correspondant aux étapes de compilation.

### 3 Options de *gcc* pour les programmes *C C++*

- **-o file** pour renommer le fichier de sortie *file*. Sans cette option, le fichier de sortie se nomme *a.out*,
- **-c -o file1.o** pour créer un fichier-objet,
- **file1.c -o a.out file2.o file3.o** pour créer un fichier exécutable résultant du programme *file1.c* utilisant des fonctions définies dans les fichiers-objet *file2.o* et *file3.o* résultats respectifs de la compilation sans édition de liens de deux programmes *C*,
- **-pipe** pour imposer l'utilisation de tubes comme médium de communication entre processus,
- **-O** réduit la taille du code et le temps d'exécution du programme,
- **-O1** augmente le code en déroulant les fonctions,
- **-O2** optimise plus que **O1**,
- **-Os** diminue au maximum la taille du programme,
- **-I répertoire** ajoute le répertoire *rep* dans la tête de liste des répertoires contenant hypothétiquement les fichiers d'entête inclus,
- **-l répertoire** ajoute un librairie dans la phase d'édition de liens,
- **-L répertoire** ajoute le répertoire *rep* dans la tête de liste des répertoires contenant hypothétiquement les bibliothèques utilisées,
- **-Dnom** réalise l'équivalent d'un **#define nom**,
- **-Dnom=valeur** réalise l'équivalent d'un **#define nom valeur**,
- **-include file** réalise l'équivalent d'un **#include file**,
- **-g** produit des informations de débogage (il annule l'option **-O**),
- **-Wall** combine toute les options **-W**,
- **-Werror** transforme tous les warnings en erreurs,
- **-pedantic** assure la norme des warnings au strict *C* et *C++ ISO*,
- **-ansi** impose au code d'être conforme à la norme *ANSI* (du langage correspondant).

### 4 Bibliothèque partagée

Une bibliothèque partagée permet de réunir un ensemble de fonctions associées à des fichiers-objet en une seule bibliothèque. *moperations.c* et *mnonoperations.c* sont deux programmes *C*. L'objectif est de créer une bibliothèque partagée regroupant les fonctions définies dans ces deux programmes.

```
gcc -c moperations.c
```

```
gcc -c mnonoperations.c
```

```
ld -G -o mabib.so moperations.o mnonoperations.o
```

Il est désormais possible d'utiliser cette nouvelle bibliothèque :

```
gcc main.c -lmabib
```