

# GCC, le jeu de compilateurs GNU



GCC (GNU Compiler Collection) est une suite de logiciels libres de compilation. On l'utilise dans le monde Linux dès que l'on veut transcrire du code source en langage machine, c'est le plus répandu des compilateurs. La suite gère le C et ses dérivés mais aussi le Java ou encore le Fortran.

Le noyau Linux dépend dans sa compilation des possibilités pointues de son compilateur C.

## 1. Installation

Il suffit d'installer le paquet **gcc** (`apt://gcc`).

À noter que lorsque l'on installe le paquet virtuel `build-essential`, il est fait appel à GCC pour la partie compilation. Mais cela ajoute d'autres utilitaires comme Make qui sont généralement nécessaires à toute compilation plus poussée que le simple bout de code. Il est donc conseillé d'installer directement **build-essential** (`apt://build-essential`).

### 1.1 Installer une autre version (...plus récente)

Si vous ne savez pas quelle version vous voulez installer, vous pouvez exécuter:

```
dpkg -l gcc*
```

Puis, exécutez, avec les droits d'administration, la liste suivante de commandes<sup>1)</sup> en remplaçant si besoin le 8 par le numéro de version voulue:

```
add-apt-repository ppa:ubuntu-toolchain-r/test
apt-get update
apt-get install gcc-8 g++-8
update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-8 60 --slave
```

Puis finalement, choisir la version de `gcc` par défaut de manière interactive :

```
update-alternatives --config gcc
```

## 2. Utilisation

## 2.1 en C

On peut appeler GCC directement par ligne de commande ou à travers un EDI ([https://fr.wikipedia.org/wiki/Environnement de développement intégré](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9)). On pourra se reporter à la page générique sur la programmation pour une liste d'EDI par langage.

Basiquement, la syntaxe est la suivante (exécution dans un terminal) :

```
gcc code.ext -o binaire
```

en remplaçant bien sûr `code.ext` par le fichier source d'entrée et `binaire` par le fichier de sortie souhaité. À défaut de paramètre *output file* (`-o binaire`), gcc créera `a.out`.

À noter qu'il faut se trouver dans le dossier où se situe le fichier.c.

## 2.2 en C++

Utilisation classique :

```
g++ fichier.cpp -std=c++11 -Wall -Wextra -o fichier_executable
```

```
-o fichier_executable
```

`-o` pour output, permet de définir le fichier de sortie. Si absent, le fichier créé est nommé `a.out`

Détermine le langage standard<sup>2)</sup>:

```
-std=c++14
```

Compilation simple (sans assemblage) : Pour simplement compiler afin de vérifier l'absence d'erreur (ou plutôt voir ses erreurs), on utilise :

```
g++ -S fichier.cpp
```

Warnings: pour que le compilateur retourne des warning sur des parties de codes qui lui semble incorrecte, on utilise généralement :

```
-Wall
```

on retrouve la liste des options que cela englobe à l'adresse ci-dessous:

<https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html> (<https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html>)

on peut compléter `-Wall` avec

```
-Wextra
```

Compilation sans édition de lien:

```
g++ -c fichier.cpp
```

à la différence de l'option `-S` vue plus haut, cela crée le(s) fichier(s) objet(s)

Compiler avec option de debuggage:

```
g++ -g fichier.cpp -o binaire_debuggable
```

Compiler avec des fichiers *include* (header files):

```
g++ -c fichier.cpp -I./include
```

Nota: vos fichier `.h` ou `.hpp` sont généralement avec vos sources, il est inutile d'ajouter cette option pour ces fichiers, en revanche, si vous avez des librairies à lier, il faut spécifier le chemin des fichiers *headers* de ces librairies.

Optimisation: `-O1` ou `-O2` ... ou `-O3`

Quelle version utilise-t-on?

```
g++ --version
```

Pour une liste exhaustive des multiples options ou optimisations et pour une syntaxe détaillée de GCC, consulter la page de manuel de gcc (longue de plus de 15000 lignes) .

## 3. Exécution

- Pour exécuter le programme/fichier binaire généré par le GCC, programmé en c par exemple, avec un fichier nommé "BinaireTest"

```
./BinaireTest
```

- Si vous avez des erreur essayez de modifier le "droit d'accès" du fichier avec cette commande :

```
sudo chmod 770 BinaireTest
```

## 4. Liens

- Site officiel : GCC, the GNU Compiler Collection (<http://gcc.gnu.org>) [en]
- Wikipedia : Page sur GCC ([https://fr.wikipedia.org/wiki/GNU\\_Compiler\\_Collection](https://fr.wikipedia.org/wiki/GNU_Compiler_Collection)) [fr]
- Tutoriel gcc : compilez sous gnu linux (<https://user.oc-static.com/pdf/31992-compilez-sous-gnu-linux.pdf>) (document non mis à jour depuis 2013)

- Débugage avec gdb



---

*Contributeurs : Johndescs, bcag2, bensuperpc*

---

<sup>1)</sup> <http://askubuntu.com/questions/466651/how-do-i-use-the-latest-gcc-on-ubuntu/581497>  
(<http://askubuntu.com/questions/466651/how-do-i-use-the-latest-gcc-on-ubuntu/581497>)

<sup>2)</sup> <https://isocpp.org/std/the-standard> (<https://isocpp.org/std/the-standard>)

 gcc.txt  Dernière modification: Le 06/02/2020, 21:38 par bcag2