



# INFO2009 - Introduction à l'informatique

## Comment installer et utiliser GCC

Bertrand Alexis

Université de Liège

2025



# Contact

## Mail

alexis.bertrand@uliege.be

## Bureau

1.81a

Institut Montefiore (b28)



# Table des matières

## Installation de GCC

Windows

MacOS

Linux

## Écrire son premier programme

## Compiler avec GCC et exécuter son programme



## Avant de commencer

**Avant de commencer à installer quoi que ce soit**, vérifie que GCC n'est pas déjà présent sur ta machine :

Ouvre un terminal et tape `gcc --version`. Si la commande te retourne la version actuelle de GCC (ou de Clang), tu peux déjà compiler tes codes C. Sinon, tu peux trouver dans ce tutoriel comment installer GCC en fonction de ton système d'exploitation.

### Ouvrir un terminal

- ▶ Windows : Le terminal peut être lancé en allant dans la barre de recherche de Windows et en entrant *cmd*.
- ▶ MacOS : Le terminal se trouve dans tes *Applications*, il se situe initialement dans le dossier *Utilities* (fr: *Utilitaires*).
- ▶ Linux : Le terminal est accessible en effectuant le raccourci clavier *Ctrl+Alt+T*.



# Installer GCC sur Windows (1/3)

Si jamais tu ne parviens pas à installer wsl ou que tu as des questions, tu peux m'envoyer un mail à [alexis.bertrand@uliege.be](mailto:alexis.bertrand@uliege.be)

## Prérequis de wsl

- ▶ Ouvre l'invite de commandes PowerShell avec les droits administrateur en allant dans la barre de recherche de Windows et en entrant **cmd** puis en appuyant sur **Exécuter en tant qu'administrateur**
- ▶ Entre la commande suivante (en 1 seule ligne)
  - ▶ `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all`
- ▶ Une fois l'opération complétée, redémarre ta machine



# Installer GCC sur Windows (2/3)

## Installer wsl

- ▶ Ouvre l'invite de commande (sans les droits administrateurs) et entre les commandes suivantes dans PowerShell :
  - ▶ `wsl --set-default-version 1`
  - ▶ `wsl --install -d Ubuntu`
- ▶ wsl te demandera un nom d'utilisateur et un mot de passe. Il est normal que tu ne vois pas ton mot de passe s'écrire sur ton terminal pendant que tu l'écris
- ▶ Une fois que tous les paquets sont installés, redémarre ta machine

## Remarque

Si la commande n'est pas reconnue, va voir la slide 8



# Installer GCC sur Windows (3/3)

## Installer GCC

- ▶ Ouvre l'invite de commandes PowerShell
- ▶ Tape `wsl` dans le terminal PowerShell pour démarrer le sous-système Linux
- ▶ Tape les commandes suivantes :
  - ▶ `sudo apt update && sudo apt upgrade`
  - ▶ `sudo apt install build-essential`
- ▶ Vérifie que GCC est correctement installé en faisant :  
`gcc --version`

## Attention

Tu peux désormais utiliser les commandes GCC pour compiler ton code C. Tu devras cependant toujours être dans `wsl` pour le faire !



# Installer wsl troubleshooting

Si tu es ici, c'est que la commande `wsl --install` n'est pas reconnue. C'est probablement parce que ta version de Windows n'a pas wsl de base. Il va falloir l'installer manuellement.

## Installer wsl manuellement

- ▶ Va sur Microsoft Store et télécharge Ubuntu
- ▶ Une fois l'installation terminée, ouvre l'application que tu viens de télécharger. Elle va s'installer puis te demander un nom d'utilisateur puis un mot de passe (c'est normal si tu ne vois pas ton mot de passe s'écrire sur ton terminal pendant que tu écris)

Tu peux maintenant avoir accès à wsl via un terminal Powershell ou celui de l'application Ubuntu et peux installer gcc !





# Installer Clang sur MacOS (1/3)

## Installer XCode

- ▶ Va sur l'App Store
- ▶ Télécharge XCode

XCode est un environnement de développement pour MacOS. Il contient notamment le compilateur Clang.

## Aide

Tu peux facilement trouver quelque chose sur ta machine grâce à *Spotlight* avec les touches **Cmd+Espace**.



## Installer Clang sur MacOS (2/3)

Maintenant que XCode est téléchargé sur ta machine, nous allons installer l'outil de ligne de commande afin de pouvoir accéder au compilateurs de XCode en dehors de ce dernier.

### Installer l'outil de ligne de commande

- ▶ Ouvre un nouveau terminal (tu peux le trouver dans tes applications ou avec Spotlight)
- ▶ Tape la commande : `xcode-select --install`



## Installer Clang sur MacOS (3/3)

Lors de l'exécution de la dernière instruction, Clang devrait avoir été installé.

- ▶ Vérifie que Clang est correctement installé en faisant :  
`gcc --version`

Tu peux désormais utiliser les commandes GCC ou Clang pour compiler ton code C !

### Remarque

Ici, nous avons installé Clang et non GCC, qui est un autre compilateur. Cependant, MacOS crée un lien entre les commandes GCC et Clang, ce qui permet de compiler avec Clang avec les mêmes instructions que GCC.



# Installer GCC sur Linux (1/2)

## Attention

Certaines distributions utilisent d'autres package managers que celui utilisé pour ce tutoriel. Si tu as une distribution qui n'utilise pas apt, n'hésite pas à venir me trouver !

## Installer GCC

- ▶ Ouvre un terminal grâce aux touches **Ctrl+Alt+T** ou en cherchant dans tes applications
- ▶ Tape les commandes suivantes :
  - ▶ `sudo apt update && sudo apt upgrade`
  - ▶ `sudo apt install build-essential`



## Installer GCC sur Linux (2/2)

GCC devrait avoir été installé avec le package *build-essential*.

- Vérifie que GCC est correctement installé en faisant :  
`gcc --version`

Tu peux désormais utiliser les commandes GCC pour compiler ton code C !

## Éditeur de texte et IDE (1/2)

Afin d'écrire ton premier programme en C, il te faudra un éditeur de texte. Les fichiers que tu écriras ont une extension particulière (.c) donc les programmes spécifiques à certaines extensions comme Word ou GoogleDoc ne permettent pas d'écrire un fichier compilable.

### Éditeur de texte

Il y a des dizaines d'éditeurs de texte et tu peux même utiliser l'éditeur de texte propre à ton OS comme *Notes* ou *TextEdit* pour écrire ton code.

Un éditeur de texte disponible pour tous les OS avec couleur syntaxique est par exemple gedit ou Sublime Text.



## Éditeur de texte et IDE (2/2)

### IDE

Les IDE (Environnement de Développement Intégré) sont des applications qui regroupent l'éditeur de texte, le débogueur, le compilateur et d'autres outils pour faciliter l'écriture de code. Nous recommandons de ne pas utiliser ce genre d'outil dans le cadre de ce cours afin que vous vous familiarisiez avec la compilation et que vous n'ayez pas d'outils facilitant l'écriture de votre code, aide que vous n'aurez pas avec vous lors de l'examen. Un IDE disponible pour tous les OS est par exemple Visual Studio Code.



## Parcourir les dossiers et fichiers (1/2)

Lorsque tu veux compiler un fichier que tu as créé, il faut que ton terminal soit ouvert dans le dossier dans lequel le fichier se trouve.

Pour ce faire, il va falloir naviguer par ton terminal dans tes dossiers. Tu peux faire ceci grâce à la commande `cd nom_du_sous-dossier`. `cd` veut dire "Change Directory" et permet donc de passer d'un dossier à un autre jusqu'à arriver au dossier dans lequel ton fichier se cache. Si jamais tu es allé trop loin, tu peux utiliser la commande `cd ..` afin de revenir au dossier précédent.

### Exemple

```
cd Documents/Introduction_informatique
```





## Parcourir les dossiers et fichiers (2/2)

Si tu ne sais pas où tu te situes, tu peux voir le chemin absolu du dossier (c'est-à dire la succession de dossiers dans lesquels tu es rentré pour y accéder) dans lequel tu es grâce à la commande `pwd`. Si tu ne sais pas quels fichiers ou dossiers il y a dans le dossier dans lequel tu te situes pour le moment, la commande `ls` va lister tous les fichiers et dossiers dans ce répertoire.

### Exemple

`ls`

- ▶ `main.c`
- ▶ `autre_dossier`

Il te suffira alors de naviguer à travers tes dossiers afin de compiler ton code C.



# Compiler un fichier

Quand tu as écrit ton code, avant de pouvoir tester ton programme, tu dois d'abord le compiler de cette manière

## Commande pour compiler

```
gcc -Wall -Wextra nom_fichier.c -o nom_fichier_exécutable
```

## Bonne pratique

Il est recommandé de toujours ajouter `-Wall` et `Wextra` lors de la compilation d'un code C. Ajouter ces flags permet de facilement repérer les erreurs de compilations et les warnings !

## Exemple

```
gcc -Wall -Wextra main.c -o HelloWorld
```



# Lancer un fichier exécutable

Une fois que tu as compilé ton ou tes fichiers C en un fichier exécutable, tu peux lancer ton programme de cette façon

## Exécuter un fichier

```
./fichier_exécutable
```

## Exemple

```
./HelloWorld
```