

Université Mohammed V
Ecole Nationale Supérieure de l'Informatique et de l'Analyse des Systèmes
Département Génie Logiciel

Environnement de programmation C

Guide de préparation

Langage C (M1.1.2)

Semestre 1 – 2026–2027

Enseignant : Youness BOUTYOUR

Séance : Préparation autonome

youness.boutyour@um5.ac.ma

Remise : Avant la première séance

Table des matières

1	Objectif du guide	2
2	Windows : VS Code et GCC avec MSYS2	2
2.1	Installer VS Code, l'extension et GCC	2
2.2	Rendre GCC disponible dans PowerShell et VS Code	3
2.3	Compiler et exécuter sous Windows	3
3	Linux : VS Code et GCC	3
3.1	Installer et vérifier les outils	3
3.2	Compiler et exécuter sous Linux	4
4	Premier programme et résultat attendu	4
5	Bonnes habitudes dans VS Code	4
6	Résoudre les problèmes fréquents	5
7	Vérification avant le TD/TP	5
	Références officielles	5

1 Objectif du guide

Ce guide permet de préparer un poste de travail pour les TD/TP de langage C. À la fin, vous devez pouvoir écrire un fichier `.c` dans Visual Studio Code (VS Code), le compiler avec `gcc`, puis exécuter le programme dans un terminal. Le débogueur `gdb` servira ensuite à observer le programme pas à pas.

VS Code facilite l'édition et le débogage, mais son extension C/C++ ne fournit pas le compilateur. Le cours utilise **C11 avec gcc**, et non C++ avec `g++`. La chaîne de travail est : **écrire** → **enregistrer** → **compiler** → **exécuter** → **vérifier le résultat**.

Outil	Rôle
VS Code	Écrire et organiser les fichiers du projet.
Extension C/C++	Aide à la saisie, repérage des erreurs et débogage.
GCC (<code>gcc</code>)	Compiler les sources C en exécutable.
GDB (<code>gdb</code>)	Examiner les variables et exécuter pas à pas.
Terminal	Lancer les commandes et le programme.

2 Windows : VS Code et GCC avec MSYS2

Ces instructions ciblent Windows 10 64 bits (version 1809 ou ultérieure) et Windows 11. Les chemins supposent l'installation de MSYS2 dans `C:\msys64` ; adaptez-les si vous avez choisi un autre dossier. [MSYS2](#) et le [guide officiel VS Code](#) décrivent cette chaîne de compilation.

2.1 Installer VS Code, l'extension et GCC

1. Installez [VS Code pour Windows](#) avec l'installateur *User Setup*. Rouvrez les terminaux après l'installation pour disposer de la commande `code`.
2. Dans VS Code, ouvrez **Extensions** avec `Ctrl+Shift+X`, cherchez **C/C++** et installez l'extension publiée par Microsoft.
3. Installez [MSYS2](#), puis ouvrez **MSYS2 UCRT64** depuis le menu Démarrer. N'utilisez pas le terminal MSYS standard pour les commandes suivantes.
4. Dans **MSYS2 UCRT64**, mettez le système à jour :

```
1 pacman -Suy
```

Si MSYS2 demande de fermer son terminal, acceptez, rouvrez **MSYS2 UCRT64** et relancez la commande jusqu'à la fin des mises à jour. Voir la [procédure officielle](#).

Installez ensuite la chaîne GCC/GDB dans ce *même terminal* :

```
1 pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain
```

Appuyez sur **Entrée** pour accepter la sélection de paquets proposée, puis confirmez l'installation. La chaîne contient notamment `gcc` et `gdb`.

2.2 Rendre GCC disponible dans PowerShell et VS Code

Dans Windows, recherchez “Modifier les variables d’environnement pour votre compte”. Modifiez la variable utilisateur Path et ajoutez une entrée contenant :

```
1 C:\msys64\ucrt64\bin
```

Ne remplacez pas ce chemin par C:\msys64\usr\bin : il désigne un autre environnement MSYS2. Fermez puis rouvrez **PowerShell et VS Code**. Dans un nouveau PowerShell :

```
1 gcc --version
2 gdb --version
3 where.exe gcc
```

La dernière commande doit pointer vers C:\msys64\ucrt64\bin\gcc.exe. Si un autre GCC apparaît en premier, vérifiez l’ordre des entrées de Path et le compilateur choisi dans VS Code.

2.3 Compiler et exécuter sous Windows

Créez un dossier personnel langage-c dans vos Documents. Ouvrez ce dossier dans VS Code avec **Fichier** → **Ouvrir un dossier**. Créez et enregistrez le fichier `bonjour.c` de la section 4. Ouvrez **Terminal** → **Nouveau terminal**, avec le profil **PowerShell**, et lancez :

```
1 gcc -std=c11 -Wall -Wextra -Wpedantic -g bonjour.c -o bonjour.exe
2 .\bonjour.exe
```

La première commande compile et crée `bonjour.exe`. La seconde l’exécute. Ces mêmes commandes fonctionnent dans un PowerShell externe ouvert dans le dossier contenant `bonjour.c`.

3 Linux : VS Code et GCC

Les commandes d’installation ci-dessous s’appliquent à **Ubuntu, Debian et leurs dérivés utilisant apt**. Pour une autre distribution, installez les paquets équivalents avec son gestionnaire de paquets. Consultez les pages officielles [installation de VS Code](#) et [GCC sous Linux](#).

3.1 Installer et vérifier les outils

Installez VS Code avec le paquet adapté à votre distribution. Dans l’éditeur, installez l’extension Microsoft **C/C++** avec **Ctrl+Shift+X**. Dans un terminal Linux, installez GCC et GDB :

```
1 sudo apt update
2 sudo apt install build-essential gdb
```

Vérifiez ensuite les trois outils :

```
1 gcc --version
2 gdb --version
3 code --version
```

Si `code` n’est pas reconnu, vérifiez l’installation de VS Code et son accès depuis le terminal. Il reste possible de lancer l’éditeur depuis le menu des applications.

3.2 Compiler et exécuter sous Linux

Dans un terminal Linux, créez et ouvrez un dossier de travail :

```
1 mkdir -p "$HOME/langage-c"
2 cd "$HOME/langage-c"
3 code .
```

Dans VS Code, créez et enregistrez `bonjour.c` selon la section 4. Ouvrez le terminal intégré et lancez :

```
1 gcc -std=c11 -Wall -Wextra -Wpedantic -g bonjour.c -o bonjour
2 ./bonjour
```

Les commandes fonctionnent également dans un terminal externe, après `cd` vers le dossier du fichier. Le préfixe `./` signifie "exécuter le fichier dans le dossier courant".

4 Premier programme et résultat attendu

Créez le fichier `bonjour.c` et enregistrez ce programme C complet :

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     /* Affiche un message puis passe a la ligne. */
6     printf("Bonjour ENSIAS !\n");
7     return 0; /* Fin normale du programme. */
8 }
```

Le terminal doit afficher exactement :

```
1 Bonjour ENSIAS !
```

L'option `-std=c11` sélectionne le standard du cours ; `-Wall -Wextra -Wpedantic` activent des avertissements utiles ; `-g` ajoute les informations de débogage. Une compilation réussie ne dispense pas de contrôler le résultat obtenu.

5 Bonnes habitudes dans VS Code

1. Ouvrez le **dossier du projet**, pas seulement un fichier `.c`.
2. Enregistrez avec `Ctrl+S` avant la compilation. GCC lit la version enregistrée sur le disque.
3. Contrôlez le dossier courant avec `Get-Location` dans PowerShell ou `pwd` sous Linux. Listez les fichiers avec `Get-ChildItem` ou `ls`.
4. Après chaque modification, **recompilez avant de relancer** : sinon vous exécuterez l'ancienne version du programme.
5. Pour l'aide à la saisie, ouvrez la palette avec `Ctrl+Shift+P`, choisissez **C/C++: Edit Configurations (UI)** et sélectionnez `gcc` et le standard **C11** si la détection automatique n'a pas retenu les bons paramètres.

6. Pour déboguer, conservez `-g`, placez un point d'arrêt, puis lancez F5 avec une configuration **GCC/GDB**. Si VS Code propose `g++`, choisissez `gcc` pour vos fichiers C.

Il permet de voir la commande réellement exécutée, les avertissements du compilateur et les entrées/sorties du programme. Évitez de dépendre d'un bouton "Run Code" ou d'un panneau de sortie non interactif avant de maîtriser **compiler**, puis **exécuter**.

6 Résoudre les problèmes fréquents

- **gcc introuvable sous Windows**

Vérifiez l'installation UCRT64, l'entrée `C:\msys64\ucrt64\bin` dans Path, puis rouvrez PowerShell et VS Code.

- **gcc introuvable sous Linux**

Vérifiez `sudo apt install build-essential` et `gcc --version`.

- **gdb introuvable**

Vérifiez la chaîne MSYS2 ou le paquet `gdb`. La compilation simple reste possible sans débogueur.

- **Fichier source introuvable**

Si GCC affiche `bonjour.c: No such file or directory`, vérifiez le dossier courant et l'enregistrement du fichier avec `Get-Location/pwd` et `Get-ChildItem/ls`.

- **L'exécutable est absent ou l'ancien résultat apparaît**

Lisez le premier message d'erreur de GCC, corrigez le fichier, enregistrez-le, puis recompilez avant de relancer.

- **Le programme ne démarre pas**

Sous Windows, utilisez `.\bonjour.exe` ; sous Linux, utilisez `./bonjour` depuis le bon dossier.

- **Le programme avec `scanf` n'accepte pas d'entrée**

Lancez-le dans un **terminal interactif**, non dans un panneau de sortie de l'éditeur.

7 Vérification avant le TD/TP

- VS Code et l'extension Microsoft C/C++ sont installés.
- `gcc --version` fonctionne dans le terminal utilisé en séance.
- `bonjour.c` est enregistré dans un dossier personnel connu.
- La compilation ne produit aucune erreur.
- Le programme affiche `Bonjour ENSIAS !`.
- Je distingue le fichier source `.c` de l'exécutable (`.exe` sous Windows).

Références officielles. VS Code sous Windows · VS Code sous Linux · GCC/MinGW avec VS Code · GCC sous Linux · Installation MSYS2 · Mise à jour MSYS2.