TP 2 : Premiers pas en programmation en langage C sous Linux

Préparé par : Dr. Ines Ben Tekaya

Objectifs du TP

À l'issue de ce TP, l'étudiant sera capable de :

1. **Comprendre l'environnement Linux pour la programmation en C**
2. **Écrire et compiler un programme C simple**
3. **Identifier et corriger les erreurs de compilation**
4. **Se familiariser avec les structures de base du langage C**
5. **Comparer différentes méthodes de compilation et d'exécution**
6. **Acquérir les bonnes pratiques de développement**

Exercice 1 : Compilation avec GCC

Dans cet exercice, nous allons utiliser un compilateur C, nommé GCC (GNU Compiler Collection) pour compiler notre premier programme écrit en C.

1. Quel est le role d'un compilateur ?
2. Pous tester l'existence de GCC, dans le terminal nous utilisons la commande :

   …….

```
vboxuser@LinuxUbuntuv1:~$ gcc --version
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Figure 1

3. En cas d'inexistence, dans le terminal taper la commande sudo apt install gcc
4. Pour installer GCC et les outils de base pour la compilation C : sudo apt install build-essential
5. Exécuter la commande de la figure 2.

```
vboxuser@LinuxUbuntuv1:~$ nano test.c
```

Figure 2

6. Ecrire le code de la figure 3

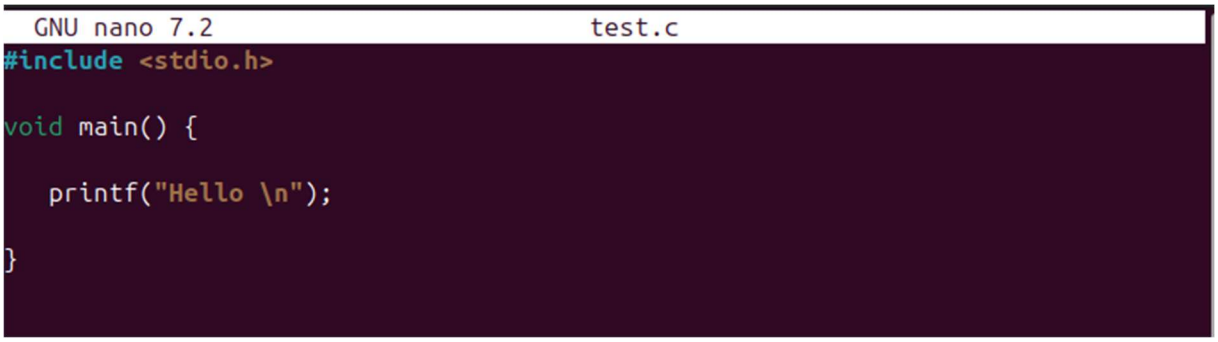

Figure 3

7. En utilisant la figure 4, exécuter la commande de compilation :

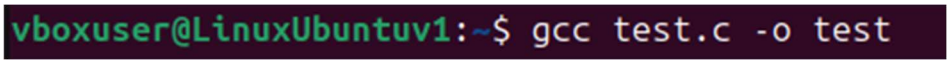

Figure 4

- Si GCC ne renvoie aucun message donc la compilation a réussi
- S'il y a des messages d'erreur, corriger le code et recompiler

8. En utilisant la figure 5, exécuter le programme :

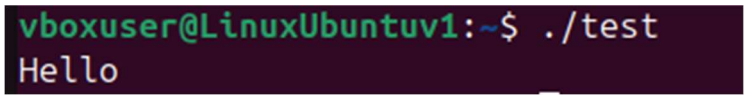

Figure 5

Exercice 2 : Compilation via un terminal intégré

Dans cet exercice, nous choisissons un IDE pour faciliter l'édition, la navigation et l'organisation du projet. L'IDE choisi est Visual Studio Code.

1. Vérifions d'abord si Visual studio code est déjà installé sur votre système. Quelle est la commande à utiliser ?
2. En cas de non existence, citer au moins deux méthodes pour installer Visual Studio Code ?
3. Choissisons la méthode la plus simple et qui se base sur le gestionnaire de paquet Snap et écrivons cette commande dans le terminal comme illustre la figure 1. Vérifions après si l'installation a bien réussi.

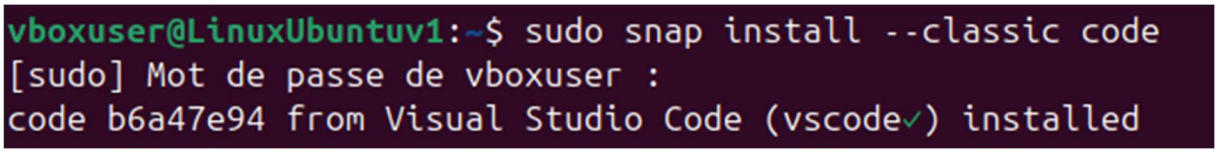

Figure 1

4. Lancer Visual Studio Code via le terminal comme illustre la figure 2.

Figure 2

5. Nous allons d'abord installer l'extension C/C++ de Microsoft comme illsutrent les figures 3 et 4.

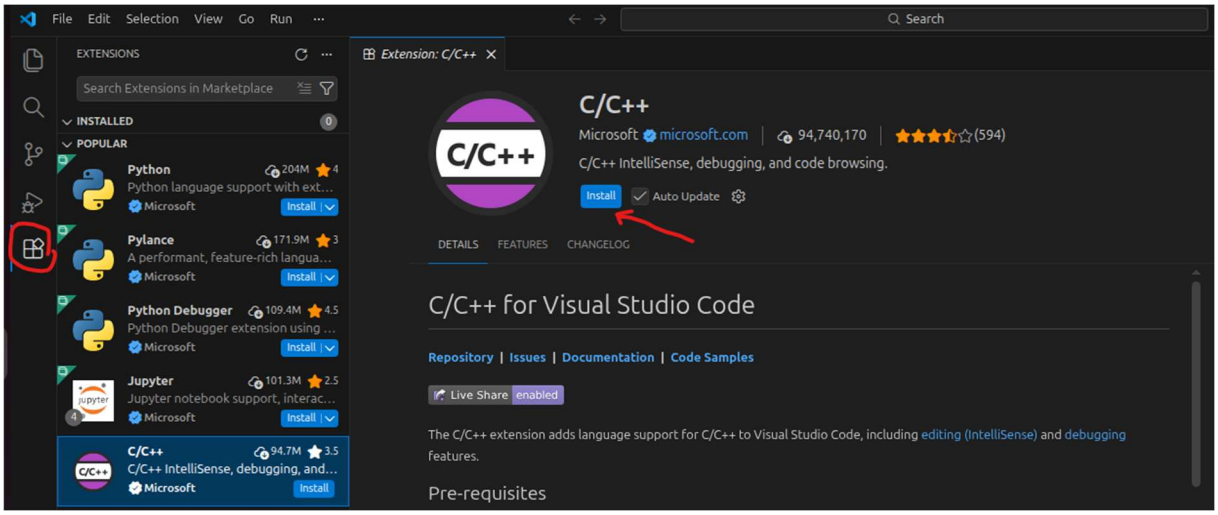

Figure 3

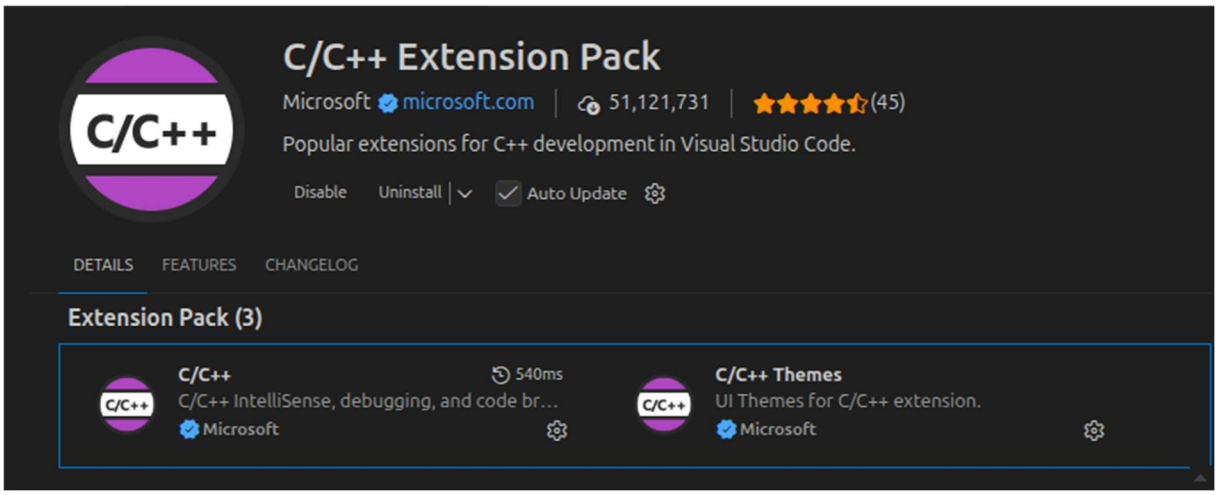

Figure 4

Pour le choix du compilateur, vous pouvez utilizer gcc qui existe déjà sous /usr/bin comme illustre la figure 5.

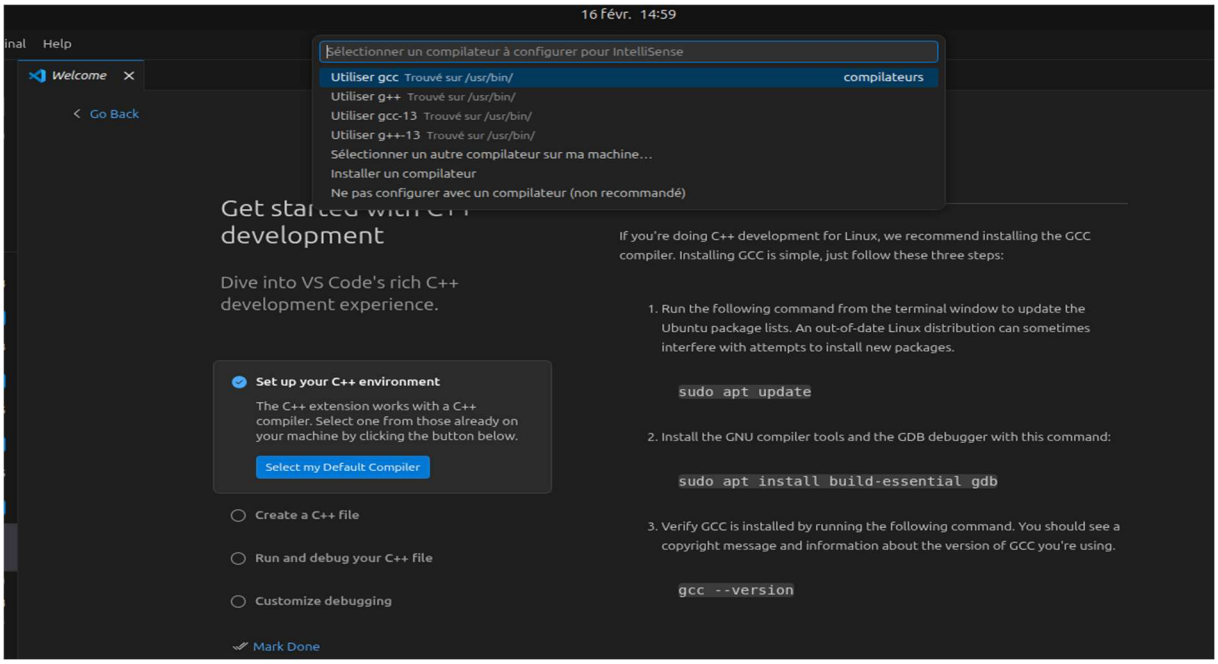

Figure 5

Commençons par créer un nouveau fichier nommé exercice2.c. Il est recommandé de **créer un dossier dédié aux TP** dans le répertoire personnel et qui se trouve sous /home/utilisateur/TPC/TP1. Les étapes sont illustrées par les figures 6, 7 et 8 pour l'utilisateur vboxuser.

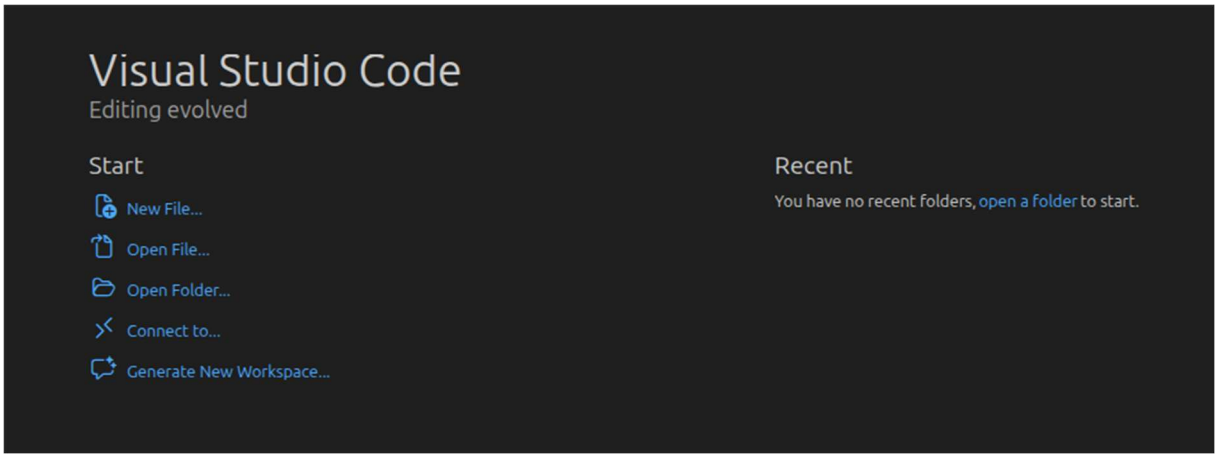

Figure 6

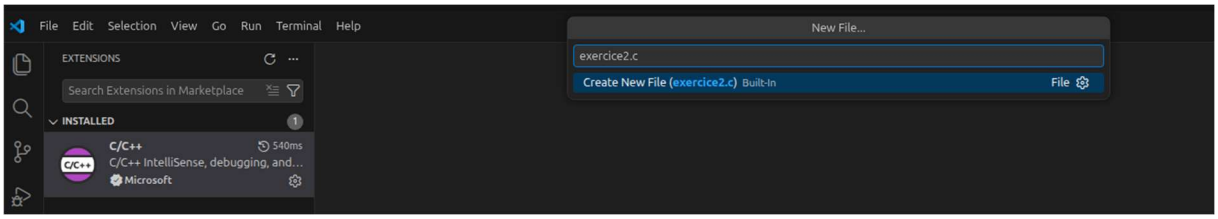

Figure 7

Figure 8

6. Exécuter ce programme qui permet d'additionner deux entiers.

```c
# include <stdio.h>
void main()
{
    int a, b;
    printf("Entrez le premier entier : ");
    scanf("%d", &a);

    printf("Entrez le deuxième entier : ");
    scanf("%d", &b);

    printf("\n ------------Résultats---------------\n");
    printf("Addition : %d + %d = %d\n", a, b, a+b) ;

}
```

7. Modifier ce programme, en affichant le résultat de la soustraction de a et b
8. Modifier ce programme en affichant le résultat de la multiplication de a et b
9. Modifier ce programme, en affichant le résultat de la division entière de a par b
10. Modifier ce programme, en permettant d'afficher le reste de la division entière de a par b.
11. Modifier ce programme en échangeant le contenu de a et b et afficher ces entiers après permutation.

Exercice 3 : Travailler avec les caractères

Écrire un programme en C qui :

1. Demande à l'utilisateur de saisir un caractère.
2. Affiche :
    - Le caractère saisi
    - Son code ASCII

Exercice 4 : Parité d'un nombre avec les structures conditionnelles

1. Ecrire un programme en C qui :

- Demande à l'utilisateur de **saisir un entier**.
- Vérifie si ce nombre est **pair ou impair**.

2. Écrire un programme en C qui calcule la valeur absolue d'un nombre saisi par l'utilisateur.

Exercice 5 : Structure itérative

Ecrivez un programme en C de deux manières et qui :

1. Demande à l'utilisateur de saisir un entier positif N.
2. Affiche tous les nombres de 1 jusqu'à N.
3. Affiche ensuite la somme de ces nombres.
4. Quelle est la méthode la plus recommandée ? expliquer.