

## UE: MU5SP05

## Lab7 : *Compilation sous GCC, Débogage*

1. Écrire une fonction **max** en C permettant de calculer la valeur max d'un tableau contenant N entier. Créez à cet effet un fichier max.c et un fichier max.h dans votre répertoire examen. Créer également un fichier testmax.c permettant de créer et d'initialiser un tableau de 5 éléments avec les valeurs suivantes: 2 5 7 2 9. Vérifier la fonction en affichant le résultat. Le prototype de la fonction est comme suit void max (int t[], int n, int \* max).

Notes : Evitez la double inclusion dans les fichiers .h via la directive.

```
#ifndef H_.....  
#define H_.....  
  
/* ----- */  
/*          */  
/* ----- */  
#endif
```

**Note :** Le fait d'avoir choisi H\_.....\_H plutôt que MAX\_H ou \_\_MAX\_H\_\_ comme drapeau n'est pas du tout le fruit du hasard. Il permet de ne pas entrer en conflit avec les identifiants réservés du langage. Par exemple, les identifiants commençant par E, LC\_, SIG, etc. sont réservés (E pour les numéros d'erreur de errno.h, LC\_ pour les constantes définies par locale.h et SIG pour les signaux de signal.h). H\_ en début d'un identifiant est libre, alors en profiter

2. Compiler, tester votre code en faisant appel au compilateur gcc. Vous pouvez faire appel au débogueur ddd pour corriger votre code (Via la commande apt-get, installer ddd (data display debugger). **Hint:** il faut compiler en mode debug avec l'option -g pour pouvoir déboguer votre code. En annexe une notice explicative du fonctionnement de ce debugger.
3. Utiliser les options -E , -S et -c pour générer les fichiers de compilations intermédiaires

4. Listez les extensions des fichiers intermédiaires générés. Expliquez l'action réalisée par le compilateur dans chacune de ces étapes.
5. Déterminer à quel niveau du process de compilation la fonction maxmin sera t'elle remplacer par le binaire correspondant.
6. Écrire une fonction **min** en C permettant de calculer la valeur min d'un tableau contenant N entier. Créez à cet effet un fichier min.c et un fichier min.h dans votre répertoire examen. Créer également dans examen un fichier testmin.c permettant de créer et d'initialiser un tableau de 5 éléments avec les valeurs suivantes: 2 5 7 2 9. Vérifier la fonction en affichant le résultat. Le prototype de la fonction est comme suit void min (int t[], int n, int \* min).
7. Écrire une fonction **variance** en C permettant de calculer la variance d'un tableau T de dimension N. Créez à cet effet un fichier var.c et un fichier var.h. Créer un fichier testvar.c permettant de créer et initialiser un tableau de 5 éléments avec les valeurs suivantes: 2 5 7 2 9. Vérifier votre fonction en affichant le résultat. Le prototype de la fonction est comme suit: void variance (int t[], int n, float \* var).

$$V = \frac{1}{N} \sum_{i=1}^N (T_i - M)^2$$

M : Moyenne des valeurs du tableau.

8. Créer un fichier test permettant de créer et initialiser un tableau de 10 éléments avec les valeurs suivantes: 4 9 10 11 12 15 12 9 0 30. Calculer et afficher le max, min et la variance.

# ANNEXE

L'interface DDD s'ouvre avec :

- Le fichier qui contient le main dans le cadre du haut
- La console de commandes gdb dans le cadre du bas
- La fenêtre de commandes ddd (command tool)

- **Afficher la fenêtre de la console**

Pour afficher la console d'exécution, sélectionner l'item Execution Window dans le menu View.

On obtient alors une fenêtre intitulée Execution window.

- **Afficher le cadre des variables**

Sélectionner l'item Data Window dans le menu View, le cadre des variables est affiché en haut de la fenêtre principale

- **Lancer l'exécution : Run**

Dans la fenêtre DDD, cliquer sur Run

Une flèche verte apparaît pour montrer l'endroit de l'exécution (sur le breakpoint).

- **Afficher la valeur d'une variable**

Un double clic sur `n_premiers_entiers` puis `somme` et `indice` dans la fenêtre contenant le code, affiche les variables et leur valeur dans le cadre du haut,

- **Se déplacer d'un breakpoint à un autre, ou pas à pas**

Pour se déplacer, il suffit de cliquer dans la fenêtre DDD sur le bouton Next pour se déplacer d'instruction en instruction, ou sur le bouton Cont pour se déplacer jusqu'au prochain breakpoint. Ce qui a pour effet d'afficher dans la console le message, et de permettre à l'utilisateur de saisir le nombre.

## Notes : Chargement de l'exécutable

Si vous modifiez l'exécutable, ou que vous voulez lancer le débogueur sur un autre exécutable, vous devez choisir l'item Open program dans le menu File, vous pouvez alors donner le nom de l'exécutable dans la fenêtre de sélection.