
Travaux pratiques 2 (version longue) : premiers pas en langage C et déboguage

Vous allez mettre tous vos programmes écrits dans ce TP dans le répertoire TP2 :

1. À partir du début de votre arborescence, créez le répertoire TP2 : `mkdir TP2`
2. Allez dans ce répertoire pour y mettre des fichiers : `cd TP2`
3. Créez un nouveau fichier source pour le langage C : `kwrite bonjour.c &`

Ce premier programme devra afficher **Bonjour !**, vous le composerez en recopiant le programme donné en exemple dans le TD, sauf les lignes 12 à 16. Gardez tous les commentaires vous en aurez besoin plus tard. Votre éditeur de texte vous assistera en colorisant automatiquement le code saisi.

Pour réaliser l’affichage vous utiliserez l’instruction : `printf("Bonjour !\n");`, dans votre fonction principale. Le `printf` est une *fonctionnalité supplémentaire* d’entrée sortie, pour qu’il fonctionne il faut insérer la ligne suivante après la ligne 2 : `#include <stdio.h> /* pour printf */`
Le `\n` représente un saut de ligne.

4. Après avoir fini d’écrire votre programme, enregistrez le.
5. Créez un programme exécutable à partir de votre fichier source, cette étape, dite de compilation, sera expliquée au prochain cours : `gcc -Wall bonjour.c -o bonjour.exe`
6. Quand l’étape précédente a réussi, exécutez le programme pour vérifier qu’il fonctionne : `bonjour.exe` (ou `./bonjour.exe`).

Vous répéterez ces trois dernières étapes (écrire/sauvegarder/compiler exécuter), très souvent ce semestre et le suivant. Exercez-vous à tout faire avec les raccourcis clavier *sans utiliser la souris*.

7. Modifiez le programme de manière à ce qu’il affiche votre prénom après le « Bonjour » (compiler/exécuter).

Partie optionnelle : déboguage

Pour ausculter vos programmes pendant leur exécution et en trouver les bugs éventuels vous pourrez faire appel à un débogueur. En salles de TP nous utilisons DDD (data display debugger). S’il vous reste du temps, voici comment faire vos premiers pas sous DDD.

8. En utilisant la fonction *enregistrer sous...* de votre éditeur de texte sauvegardez votre programme sous le nom `echange.c`, compilez le sous le nom `echange.exe` et exécutez le (`echange.exe` ou `./echange.exe`).
9. Déclarez une variable entière `x` dans votre programme et après le `printf`, affectez lui la valeur 10.

Pour que le débogueur fonctionne au mieux il faut ajouter des informations à l’exécutable, au moment de sa création, à l’aide de l’option `-g`.

10. Recommencez la compilation mais en ajoutant l’option `-g` à la commande gcc : `gcc -g -Wall echange.c`
11. Ouvrez le débogueur sur ce nouvel exécutable : `ddd echange.exe`
12. Lancez une exécution complète de votre programme (bouton *Run*).

13. Les points d'arrêt (*breakpoint*) permettent de suspendre temporairement l'exécution du programme. À l'aide d'un clic-droit poser un point d'arrêt, sur la ligne avant le *printf*. Lancez l'exécution avec *Run*. Lorsque le programme atteint le point d'arrêt, le débogueur suspend son exécution. Profitez en pour ouvrir la visualisation des variables locales (menu *Data* puis *Display local variables*). Progressez dans l'exécution du programme pas à pas à l'aide du bouton *Step*.
14. Quittez le débogueur, modifiez votre programme de manière à ce qu'il effectue l'échange des valeurs entre deux variables *x* et *y* que vous initialiserez à des valeurs de votre choix (comme en TD), et lancez l'exécution pas à pas de votre programme dans le débogueur, comme aux étapes précédentes. Cela fonctionne t'il correctement ?