

TP2 - Environnement de développement en C

A partir de ce TP nous allons programmer en langage C. La programmation dans un langage impératif nécessite la compilation du programme source pour obtenir un exécutable.

Création d'un exécutable à partir d'un programme C

Nous utilisons le compilateur gcc de GNU qui est un logiciel libre sous licence GPL, disponible gratuitement (<http://gcc.gnu.org/>). Pour créer un programme exécutable à partir d'un programme source écrit en C, une seule commande est nécessaire avec gcc pour :

1. Compiler les sources (.c) et créer les fichiers objets (.o),
2. Faire l'édition de liens, c'est à dire résoudre les références internes et externes (appels aux bibliothèques) pour produire un exécutable.

Compiler et exécuter un programme simpliste

Créer un répertoire de travail dans votre répertoire personnel. Utiliser un éditeur de texte pour taper le programme suivant. Vous pouvez utiliser KEdit (similaire au Notepad de Windows) ou Xemacs (plus complexe mais plus puissant).

```
#include <stdio.h>
void main()
{
    printf ("Bonjour !!\n");
}
```

Enregistrer le fichier sous le nom '*bonjour.c*'.

La deuxième étape est la compilation du programme. Pour cela, ouvrez un terminal, déplacez-vous dans votre répertoire de travail et tapez :

```
> gcc bonjour.c -o bonjour
```

gcc est la commande utilisée pour accéder au compilateur. La commande *-o* ordonne à *gcc* de créer un exécutable appelé *bonjour*.

Le compilateur a généré un fichier '*bonjour*' qui est la version exécutable du programme C que vous pouvez exécuter par :

```
./bonjour
```

Si tout s'est bien passé, vous devez voir s'afficher :

```
Bonjour !!
```

Éditeur de texte : [X]emacs

Xemacs est un éditeur de texte dont la particularité est de proposer un langage de programmation (elisp, un dérivé du lisp) pour ajouter des fonctionnalités à l'éditeur. Cela permet notamment de proposer des fonctions comme la coloration syntaxique d'un code source ou l'appel d'un programme externe pour compiler directement de l'éditeur.

Nous vous donnons ici le matériel de survie minimum pour Xemacs. La lettre C désigne ici la touche Control, et M (prononcer Meta) la touche alt gauche.

ouverture/fermeture de fichier : C-x C-f fichier : ouvre un fichier/C-x C-s sauve le fichier/C-x C-w fichier sauve le fichier en demandant un nom.

copier/coller : à la souris ou bien : sélection de bloc avec C-espace, copier M-w, couper C-w, coller C-y.

coloration syntaxique : après avoir ouvert un fichier C, faire M-x font-lock-mode.

compilation : en mode C : M-x compile puis choisir la commande gcc.

aide : C-h a : rentrer une chaîne de caractères

quitter : C-x C-c.

Toutes ces commandes sont également accessibles par les menus.

Fonctions d'entrée/sortie

Pour entrer des données ou les afficher à l'écran vous devez utiliser les fonctions entrée/sortie définies dans le fichier *stdio.h*. Il faut donc que votre programme commence par :

```
#include <stdio.h>
```

Précision sur les fonctions d'entrée/sortie de base :

printf(format,listevaleurs) qui affiche la liste de valeurs (variables ou expressions) dans le format choisi.

Printf retourne le nombre de caractères écrits, ou EOF en cas de problème. Faire man printf pour plus de détails.

scanf(format,listeadresse) pour la lecture au clavier de valeurs, dans le format spécifié. Dans le cas de variables scalaires, il faut les faire précéder de l'opérateur & (scanf attend un pointeur). Scanf retourne le nombre de valeurs effectivement lues et mémorisées. Les séparateurs sont le retour charriot, le blanc et le tab. On peut donc séparer par des CR ou des blancs plusieurs valeurs demandées dans le même scanf. Mais de même, si scanf a pu lire tous ses arguments sans arriver à la fin de la ligne, la suite servira au prochain scanf.

Dans le cadre des TP on utilisera les formats sous leur forme la plus simple comme par exemple :

```
printf("toto= %d\n",toto); /* toto est un int */
scanf("%d",&toto);
```

Exercice : conversion de température

Faites un programme qui demande une température en degrés Celsius ou en degrés Farenheit, et qui convertit la valeur d'une unité à l'autre (on se rappelle que $T(\text{degrés}) = 5/9 * (T(\text{Farenheit}) - 32)$).

Exercice : factorielle

Faites un programme qui demande un nombre n , puis calcule la factorielle de ce nombre en utilisant

1. une variable de type int
2. une variable de type unsigned int
3. une variable de type unsigned long
4. une variable de type float

Constater que C ne donne pas de message d'erreur en cas de dépassement de capacité.

Exercice : calcul sur les nombres

Ecrire en C le programme vu en TD qui lit N nombres entiers et calcule la moyenne (float), le minimum et le maximum. On utilisera **une seule boucle** pour tous les calculs et **une seule variable** pour lire les entiers. Le nombre N sera aussi demandé à l'utilisateur.

Exercice : calcul de π

On utilisera la formule de Gregory : $\frac{\pi}{4} = \sum_{0 \leq n \leq K} \frac{(-1)^n}{2n+1}$

Faire un programme itératif qui calcule π avec deux arrêts possibles :

- Le programme demande la précision (Ex. 10^{-2}) et s'arrête quand cette précision est atteinte en affichant le calcul approché de π et le nombre d'itérations,
- Le programme demande le nombre d'itération et s'arrête quand ce nombre est atteint en affichant le calcul approché de π et la précision entre les deux derniers termes calculés.

On donnera à l'utilisateur le choix de la méthode.

Remarque : La série a une croissance très lente.