

TP 9 : Préprocesseurs et macros

Programmation en C (LC4)

Semaine du 24 mars 2008

1 Utilisation du préprocesseur

Exercice 1 Ecrire les macros suivantes :

- Définissant une constante PI
- Calculant le cube d'un nombre
- Calculant la somme de deux nombres
- Calculant le volume d'une sphère ($V = (4\pi R^3)/3$).

Vous utiliserez ces macros dans un main permettant d'afficher la somme des volumes de N sphères, dont le rayon est passé en paramètre du programme.

Vous pourrez utiliser la fonction double `atof(char * chaine)` qui convertit une chaîne de caractère en nombre flottant.

2 Compilation conditionnelle

Exercice 2 La compilation conditionnelle permet d'orienter la compilation suivant certains critères (compilation sous différentes plateformes par exemple), ici nous allons voir comment remédier aux dépendances cycliques entre modules grâce à la compilation conditionnelle.

- Créez cinq fichiers `tata.h`, `tata.c`, `toto.h`, `toto.c`, et `main.c`.
- Dans `toto.h` sera défini la fonction `somme` qui calcule la somme de deux entiers, cette fonction sera réalisée dans `toto.c`.
- Dans `tata.h` sera défini la fonction `produit` qui calcule le produit de deux entiers, cette fonction sera réalisée dans `tata.c`.
- Le fichier `tata.h` doit dépendre du fichier `toto.h` et réciproquement.
- le fichier `main.c` appellera ces deux fonctions.
- Ecrivez la Makfile qui permet de compiler l'ensemble de ces fichiers. Que se passe-t-il ?
- Remédiez à cette erreur en utilisant la compilation conditionnelle.

3 Programmation modulaire

Exercice 3 Le but de cette exercice est de masquer l'implémentation d'un vecteur et de n'utiliser que les fonctions définies dans l'interface.

- Définir dans un fichier `type.h` un type vecteur composé de trois réels, ainsi que les fonctions `creer_vecteur` qui crée un vecteur, et `coord_1`, `coord_2`, `coord_3` qui renvoient respectivement la première, deuxième et troisième coordonnée du vecteur qui leur est passé en paramètre. Ces fonctions seront réalisées dans le fichier `type.c`
- Définir dans un fichier `norme.h` l'en-tête d'une fonction `norme` prenant en argument un vecteur et renvoyant sa norme. Cette fonction sera réalisée dans le fichier `norme.c`.
- Définir dans un fichier `produits.h` les en-têtes des fonctions `produit_scalaire` et `produit_vectoriel` prenant en argument deux vecteurs et renvoyant respectivement leur produit scalaire et vectoriel. Ces fonctions seront réalisées dans le fichier `produits.c`.

- Notons les vecteurs $x = (x_1, x_2, x_3)$ et $y = (y_1, y_2, y_3)$

- Norme : $\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$

- Produit scalaire : $x \cdot y = x_1 * y_1 + x_2 * y_2 + x_3 * y_3$

- Produit vectoriel : $x \wedge y = (x_2 * y_3 - x_3 * y_2, x_3 * y_1 - x_1 * y_3, x_1 * y_2 - x_2 * y_1)$

- Ecrire dans un fichier `combinaison.c` un programme qui à partir de deux vecteurs et qui calcule la norme de leur produit vectoriel.
- Ecrire un `Makefile` qui permet de compiler l'ensemble de ces fichiers.