

Pratique de la programmation et projet

TP 2 : préprocesseur, compilation séparée et bibliothèques

Frédéric Vivien

Utilisation du préprocesseur

1. Écrivez des macros :

- (a) Définissant une constante `N` valant 25 ;
- (b) Définissant une constante `M` égale au carré de `N` ;
- (c) Calculant la valeur absolue d'un nombre ;
- (d) Calculant le maximum de deux nombres.

Vous utiliserez toutes ces macros dans un programme qui tire au sort `M` nombres (en utilisant la fonction `random`) et qui retourne le nombre dont la valeur absolue est maximale.

2. Inclusion de fichiers en cascade :

- (a) Définissez, dans un fichier `type.h`, un type `vecteur` composé de trois réels.
- (b) Définissez, dans un fichier `norme.h`, l'en-tête d'une fonction `norme` prenant en entrée un vecteur et renvoyant sa norme. La fonction `norme` sera réalisée dans le fichier `norme.c`.
- (c) Définissez, dans un fichier `produits.h`, les en-têtes des fonctions `scalaire` et `vectoriel` prenant deux vecteurs en entrée et renvoyant respectivement leur produit scalaire et leur produit vectoriel. Ces fonctions seront réalisées dans le fichier `produits.c`.

$$\text{Produit scalaire : } \begin{vmatrix} a \\ b \\ c \end{vmatrix} \cdot \begin{vmatrix} x \\ y \\ z \end{vmatrix} = ax + by + cz, \quad \text{produit vectoriel : } \begin{vmatrix} a & x \\ b & y \\ c & z \end{vmatrix} = \begin{vmatrix} bz - cy \\ cx - az \\ ay - bx \end{vmatrix} .$$

- (d) Réalisez dans un fichier `combinaison.c` un programme qui saisit deux vecteurs et calcule la norme de leur produit vectoriel.

Génération d'une bibliothèque

À partir des fichiers sources `norme.c` et `produits.c` de la question précédente on veut créer une bibliothèque (ou archive) `vecteur`.

1. statique ;
2. dynamique.

Dans les deux cas vous générerez l'exécutable `combinaison` au moyen de ces bibliothèques.

Vous regarderez les fichiers `norme.o`, `produits.o`, `libvecteur.so`, `combinaison.o` et `combinaison` au moyen de la commande `nm`.

Compilation séparée

1. Dans l'exercice précédent tous les fichiers d'en-têtes et tous les fichiers sources étaient dans le même répertoire. Mettez les fichiers sources dans un même répertoire et les trois fichiers d'en-têtes dans trois répertoires différents. Compilez le tout (en utilisant l'option `-I` de `gcc`).

2. Vous trouverez sur la page <http://icps.u-strasbg.fr/~vivien/Enseignement/PPP-2001-2002/> une archive contenant les sources d'un programme (`sources-exemple.tgz`). Pour pouvoir compiler ces sources et obtenir un programme `imscan` vous devrez utiliser deux bibliothèques :
 - (a) La bibliothèque multi-précision de GNU, renommée dans notre cas `magmp` et installée sur `ada` dans le répertoire : `/users/prof/vivien/IciEstLaGmp`.
 - (b) La bibliothèque `polylibgmp`, installée sur `ada` dans le répertoire : `/users/prof/vivien/LaPolylib`, et qui a besoin de la bibliothèque précédente.

Vous compilerez ce programme en faisant une édition de liens

- (a) statique ;
- (b) dynamique.

Dans les deux cas vous testerez votre programme en l'exécutant sur le fichier `imscan-exemple` (`imscan` ; `imscan-exemple`).

Quels sont les problèmes que vous rencontrez ? L'ordre dans lequel les bibliothèques sont passées au compilateur est-il important ? Dans les deux cas, pensez à regarder la taille des exécutables générés.