Architecture des ordinateurs

L3 Informatique 2013-2014

TD/TP 8 - Assembleur et C

Points abordés

- Encodage des instructions.
- Écrire des fonctions en assembleur dans un code C.

Exercice 1. (Encodage d'une instruction)

Écrivez un exemple simple de programme assembleur utilisant l'instruction `inc eax` et ouvrez l'exécutable obtenu sous DDD.

(a) Identifiez le code en langage machine qui correspond à `inc eax`.

(b) En testant de même les différents arguments de l'instruction `inc`, déterminer la manière dont cette instruction est codée en langage machine.

Exercice 2. (Renverser un tableau)

Écrivez un programme C, qui crée un tableau d'entiers de n cases et le remplit comme vous le souhaitez, puis le renverse (c'est à dire échange les cases i et $n-i-1$) à l'aide d'une fonction `Renverse(tab,n)` (où *tab* est le tableau et n sa taille) qui sera codée en assembleur. Votre fonction ne devra pas utiliser de tableau annexe.

Exercice 3. (Crible d'Ératosthène)

Le crible d'Ératosthène est une méthode rapide pour déterminer quels sont les entiers premiers plus petit que n. Il consiste à initialiser un tableau *tab* de n cases en mettant toutes ses valeur à 1, sauf les cases 0 et 1 qui sont mises à 0. Ensuite, pour chaque case $i \geq 2$, si $tab[i]$ vaut 0, on ne fait rien et s'il vaut 1, on met à 0 les cases $k \times i$ avec $k > 1$ et $k \times i < n$.

Écrivez un programme en C qui implémente le crible d'Érathosthène. Le programme principal, la réservation mémoire et les entrées-sorties seront faits en C et la fonction calculant le crible sera faite en assembleur. Cette fonction prendra en argument la taille du tableau.

Exercice 4. (Créer un flottant)

1. Écrire un programme en C qui vous demande trois entiers s, e et m, puis qui crée un flottant f dont la représentation en binaire est $s.(e+127).m$ (comme vu au TP 2 et 3), à l'aide d'une fonction en assembleur `int_to_float(s,e,m)`. On rappelle que s vaut 0 ou 1, e est compris entre -127 et 128 et m vaut au plus 2^{23}.
2. Ajoutez maintenant une fonction en assembleur `float_to_int(f,&s,&e,&m)` qui prend en argument un flottant f et trois pointeurs vers des entiers s, e et m et stocke dans ces derniers la décomposition de la représentatation de f en signe, exposant et mantisse.