

Assembleur x86 TP n°1

A. Appel de fonctions

- a. Ouvrez le projet HelloWorld donné en exemple puis le compiler à l'aide du fichier make.bat fourni.
- b. Exécutez ce programme à l'aide du débogueur et regardez comment est réalisé l'appel à la fonction.
- c. Etudier pas à pas l'évolution de la pile pendant l'exécution du programme.
- d. Créez un nouveau projet avec un programme écrit en assembleur réalisant l'appel à MessageBox.

B. Modes d'adressage

- a. Réalisez une routine pour mettre en majuscule une chaîne de caractères.
- b. Placez cette routine dans des sous-programmes que vous appellerez en passant l'adresse de la chaîne de caractères par la pile.
- c. Réalisez un sous-programmes permettant de compter les caractères d'une chaîne

C. Variables locales

- a. Traduire la fonction C suivante en assembleur (utiliser des variables locales pour j, k et l) :

```
int myst( int n ){
    int i, j, k, l;

    j = 1;
    k = 1;
    for ( i = 3; i <= n; i++ ) {
        l = j + k;
        j = k;
        k = l;
    }
    return k;
}
```

- b. Que calcule cette fonction ? Vérifier que le programme assembleur donne bien le résultat attendu.

- c. Ecrire et tester une fonction qui, étant donné un mot (une chaîne de caractères) sur l'alphabet {a, b, c}, renvoie le nombre de 'a' (respectivement de b, de c) dans eax (respectivement dans ebx, dans ecx). Utiliser des variables locales pour compter les lettres.

D. Un peu de calcul

- a. Ecrire un programme qui lit un entier positif au clavier et affiche tous les entiers qui divisent ce nombre. Par exemple si le nombre lu est 10, le programme affichera : 1 2 5 10.
- b. Ecrire une fonction récursive qui calcule la factorielle d'un entier lu au clavier.

E. Un peu de lecture

- a. Lire et analyser l'article suivant dans le but d'en faire un résumé succinct :

<http://www.segmentationfault.fr/reversing/plongeon-dans-les-appels-systemes-windows/>