

# Travaux Dirigés de Compilation n°4

## Licence d'informatique

---

### Fonctions en assembleur

Le but de ce TD est d'apprendre à implémenter des fonctions en assembleur et à se familiariser avec les conventions AMD 64.

---

#### ► Exercice 1. *Arguments et valeur de retour*

1. Écrivez en assembleur une fonction qui prend un entier signé et qui renvoie son carré. Respectez les conventions d'appel AMD 64 pour passer le paramètre et renvoyer la valeur. Votre code doit pouvoir traiter des entiers signés codés sur 4 octets.
2. Écrivez dans un autre fichier du code en assembleur qui appelle votre fonction sur l'entier de votre choix, et qui affiche le résultat.

#### ► Exercice 2. *Registres volatils et non volatils*

1. Écrivez en assembleur un programme qui alloue de la mémoire statique pour deux entiers signés de votre choix, sur 4 octets chacun. Écrivez en assembleur une fonction qui prend en paramètre deux adresses, et qui intervertit les données (de taille 4 octets) à ces adresses, comme dans la fonction C suivante :

```
void swap(int *a,int *b){  
    int c=*a;  
    *a=*b;  
    *b=c;  
}
```

Respectez les conventions d'appel, y compris en ce qui concerne les registres volatils et non volatils. Testez en appelant votre fonction et en affichant les résultats.

2. Insérez des appels à `show_stack`
  - avant l'appel à votre fonction
  - et au milieu de votre fonction.

#### ► Exercice 3. *Alignement de la pile*

1. Écrivez en assembleur une fonction récursive qui prend en arguments deux entiers positifs ou nuls  $m$  et  $n$  et qui renvoie  $mn$  en utilisant `add` mais ni `imul` ni `mul`. Respectez les conventions d'appel, y compris en ce qui concerne l'alignement de la pile. Testez en appelant votre fonction sur deux entiers positifs ou nuls de votre choix, et en affichant les résultats.
2. Insérez dans votre fonction récursive un appel à `show_stack` avant l'appel récursif. Respectez les conventions d'appel.