



# TP Assembleur x86

## Calculatrice simple

### 1 Objectif du TP

Realiser progressivement une calculatrice simple en assembleur x86 (addition, soustraction, multiplication, division).

### 2 Programme final attendu

Le programme final devra :

- afficher un menu (choix de l'operation),
- lire deux nombres (un chiffre de 0 a 9),
- effectuer l'operation choisie,
- afficher le resultat,
- terminer proprement.

### Bareme global

- Phase 1 : 2 points
- Phase 2 : 3 points
- Phase 3 : 3 points
- Phase 4 : 4 points
- Phase 5 : 4 points
- Phase 6 : 4 points

### 3 Decoupage du TP en etapes

#### 3.1 Phase 1 : Affichage d un message (2 points)

**Consigne :** Ecrire un programme assembleur qui affiche un message puis se termine.

**QCM :**

- Quelle section contient les instructions ?
- Quel registre contient le numero de l'appel systeme ?

#### 3.2 Phase 2 : Lecture clavier (3 points)

**Consigne :** Lire un caractere depuis l'entree standard et l'afficher.

**QCM :**

- Quel est le numero de sys\_read ?
- A quoi sert ECX lors de la lecture ?

#### 3.3 Phase 3 : Addition simple (3 points)

**Consigne :** Lire deux chiffres ASCII, les convertir en nombres et afficher leur somme (inferieure a 10).

**QCM :**

- Pourquoi soustrait on '0' ?
- Quel registre contient le resultat ?

#### 3.4 Phase 4 : Menu et CMP / JE (4 points)

**Consigne :** Afficher un menu et executer une action selon le choix utilisateur.

**QCM :**

- Que fait CMP ?
- Quand est ce que le "JE" saute t il ?

#### 3.5 Phase 5 : Multiplication et division (4 points)

**Consigne :** Utiliser MUL et DIV et expliquer le role de AL, AH et AX.

**QCM :**

- Ou se trouve le reste apres DIV ?
- Pourquoi faut il nettoyer AH ?

#### 3.6 Phase 6 : Programme complet (4 points)

Assembler toutes les phases pour obtenir une calculatrice fonctionnelle.

**Consigne :** Afficher un menu, lire le choix utilisateur et executer l'operation correspondante.

**QCM :** Quelle instruction permet de comparer deux valeurs ?

- a) MOV
- b) CMP

## 4 Commandes de compilation

```
nasm -f elf32 calc.asm -o calc.o  
ld -m elf_i386 calc.o -o calc  
./calc
```