

## TP 1

### Utilisation du simulateur Motorola 68000

Pour lancer le simulateur Motorola 68000, il faut copier le fichier dont le chemin est précisé ci-dessous dans votre répertoire de travail :

```
/home/commun_depinfo/enseignants/boukhobza/sim68.jar
```

### Utilisation du simulateur

Les étapes à suivre :

Etape 1. Taper le programme source à l'aide d'un éditeur de texte de votre choix.

```
DATA
X:      DC.L    $aabbccdd
Y:      DC.B    5
TAB:    DS.B    10

CODE
MOVE.B  X, D0
MOVE.L  X, D1
MOVE.B  Y, D2
BCLE:   ADD.L   #1, D1
        SUB.B   #1, D2
        BNE    BCLE
RTS
```

Etape 2. Sélectionner le programme source depuis le simulateur (commande *Source*)

Etape 3. Assembler le programme source (commande *Assembler*)

Etape 4. Charger le programme (commande *Charger*)

Etape 5. Exécuter le programme (commande *Exécuter*)

*Note* : après une modification du programme source, il suffit de reprendre les étapes 3 à 5. Il n'est pas nécessaire de refaire "source" chaque fois qu'on modifie un programme. Il suffit de faire "assembler".

*Note* : pour ré-exécuter un programme, il suffit de reprendre les étapes 4 et 5.

### Éléments de programmation en langage d'assemblage

Les constantes

```
DATA
MAX EQU 10
```

Les fonctions

```
CODE
JSR  INIT
```

```

        JSR     CALCUL
        RTS
INIT:   MOVE.L #MAX,D1
        RTS
CALCUL:ADD.L #2,D2
        RTS

```

## Exercice 1

Programmer les fonctions suivantes ainsi que des exemples d'appels de ces fonctions :

1. if ((d0 > 1) && (d0 < 4)) d1 = 4;
2. if ((d0 <= 1) || (d0 >= 4)) d1 = 5;
3. switch (d0) {
  - case 1: d1=4; break;
  - case 2: d1=5; break;
  - default: d1=6;
4. while (d1 < 2) {d0 = d0+d2; d1 = d1+1}
5. do {d2 = d0+d2;} while (d2 < 10);
6. for (d7=1 ; d7<d6 ; d7++) {d2=d0+d2;}

*Note:* on pourra utiliser l'instruction CMP pour effectuer les comparaisons. L'instruction CMP est semblable à l'instruction SUB mais le résultat n'est pas conservé (on conserve quand même les informations sur le signe du résultat, ce qui permet d'utiliser des instructions du type BEQ).

## Exercice 2

Ecrire une fonction qui initialise un tableau d'octets de la manière suivante :

```

TAB[0] = 'D';
TAB[1] = 'E';
TAB[2] = 'F'; ...

```

Paramètres : adresse du tableau dans A1, nombre d'éléments du tableau dans D1, valeur du premier élément du tableau dans D2.

Ecrire une fonction qui indique si un caractère est présent dans un tableau.

Paramètres : adresse du tableau dans A1, taille du tableau dans D1, caractère à rechercher dans D2.

Résultat : valeur 1 dans D0 si trouvé, valeur 0 dans D0 si non trouvé.

## Exercice 3

Mêmes questions que dans l'exercice 2 avec un tableau de mots longs.

```

TAB1[0] = 3;          TAB2[0] = 3;
TAB1[1] = 4;          TAB2[1] = 4;
TAB1[2] = 5;          TAB2[2] = 5;

```

## Exercice 4

1. Ecrire une fonction qui calcule la somme des N premiers entiers. Paramètre : valeur de N dans D7. Résultat dans D0.
2. Ecrire une fonction qui calcule la somme des N premiers entiers pour toutes les valeurs de N comprises entre 1 et M et range les résultats dans un tableau. Paramètres : valeur de M dans D1, adresse du tableau dans A1.