TD n° 4 : FONCTIONS ET PROCEDURES

1. Introduction

La première partie du TD-TP utilise le jeu d'instructions ARM et ARMSIM.
La seconde partie (***travail personnel***) utilise le jeu d'instructions MIPS et QtSpim

2. Ecriture de programmes assembleurs ARM avec procédures

A. *Procédures simples*

a) *MAX de deux variables*

```
int a, b, c ;
main(){
B=10; C=15;
a= max (b,c) ;
}

int max (int x, int y) {
if (x > y) return x ;
      else return y;
}
```

Le programme TD4L3A1.s est une version ARM du programme C avec passage des paramètres par registres.
Le programme TD4L3A2.s est la version du programme avec passage des paramètres par la pile.
Exécuter les deux programmes en observant le passage des paramètres et du résultat.

b) *Fonction calculant la somme des valeurs d'un tableau d'entiers*

Compléter les deux programmes TD4L3A3.s et TD4L3A4.s utilisant une procédure pour calculer la somme d'un tableau d'entiers signés (la version sans procédure a été vue dans le TD3).

- **TD4L3A3.s** passera les paramètres par registres
- **TD4L3A4.s** passera les paramètres par la pile.

B. *Procédures imbriquées*

Soit le programme C suivant

```
int foo(int i) {
    return (i+1);
}
int bar (int i) {
    int k;
    k=foo(i);
    return (i+k);
}

main () {
int i =2;
int j;
j=bar (i);
  }
```

Ecrire les procédures foo et bar et le programme appelant en passant les paramètres par registres (TD4L3A5.s)

3. Programmes assembleur ARM générés par un compilateur

Conventions du compilateur

- Les quatre premiers arguments sont passés par R0-R3 dans cet ordre. S'il y a plus d'arguments, ils sont passés par la pile.
- La valeur de retour passe par R0
- L'appelée peut modifier le contenu des registres R0-R3 et R12 : une sauvegarde éventuelle est à la charge de l'appelante.

A. Procédures imbriquées

Le programme TD4L3A6.s est la version assembleur générée par un compilateur gcc pour le programme C de la question 2.B.
Exécuter le programme en observant notamment l'évolution de la pile.
On observera notamment

- La manière dont sont passés les paramètres
- Les paramètres sauvegardés par les procédures appelantes
- L'utilisation du pointeur de trame (frame pointer).

Peut-on optimiser le code de foo en conservant les conventions du compilateur ?

B. Procédures récursives

Le programme TD4L3A7.s est la version assembleur générée par un compilateur gcc du programme C ci-dessous (version récursive du calcul du $n^{ième}$ élément de la suite de Fibonnaci).

```
int fibo1(int a) {
    if (a <= 1) return a;
    return (fibo1(a-1) + fibo1(a-2));
}
main () {
int n;
n=fibo1(8);
    }
```

Exécuter le programme en observant notamment l'évolution de la pile.

- Quelle est la taille totale de l'espace mémoire utilisé ?
- Pourquoi le registre r4 est-il sauvegardé ?
- Comment est passé le paramètre de fibo ?

Optimiser ce programme en respectant les conventions du compilateur.

4. Jeu d'instructions MIPS (travail personnel)

- Ecrire un programme **TD4L3M1.s** qui calcule la somme des N entiers signés d'un tableau en utilisant une procédure (passage des paramètres par registres)
- Le programme TD4L3A8.s calcule la somme des N premiers entiers de manière récursive. Ecrire la version MIPS **TD4L3M2.s** de ce programme.