IUT Béziers
Département Réseaux et Télécoms
B.U.T. R&T

2021/2022
R106
S.D.

R106 – Architecture des ordinateurs

Assembleur 1/2

TP 2

Consignes

Comme pour le TP précédant, ce compte-rendu de TP vous servira a préparer l'examen pratique final. Vous pouvez le faire corriger par un enseignant si vous le souhaitez.

Ce TP utilise un simulateur ARM disponible en ligne à l'adresse :

http://gif1001-sim.gel.ulaval.ca/?sim=nouveau

Vous aurez également besoin du set d'instructions des processeurs ARM, disponible en ligne sur le site du cours.

1 Conseils de rédaction

Suite à la lecture de quelques comptes rendu lors de la dernière séance, il me semble nécessaire de rappeler quelques bonnes habitudes.

- **Un bon compte rendu doit être clair et concis**. Si vous devez relire une demi page avant de localiser la bonne information, vous vous y prenez mal.
- **Les commandes doivent être notées précisément et apparaitre clairement.** Évitez de "noyer" une commande dans un paragraphe de texte. Notez la sur une ligne dédiée et ne faites pas d'erreur en la recopiant !
- **Répétez la question** puis répondez-y. Un compte-rendu doit se suffire à lui même, c'est à dire qu'il peut être lu sans disposer de l'énoncé.
- **Ne pipeautez pas.** Cela ne sert à rien : Ce compte-rendu est pour vous !

2 Premiers pas ...

Exercice 1 Testez le programme étudié ce matin en TD. Fonctionne-t-il ?
Exécutez le pas à pas.

Exercice 2 Écrivez un programme inscrivant la valeur 0xABBA (sur 32bits) à l'adresse 0x842 en mémoire.

Exercice 3 **Copie de mémoire**

Écrivez un programme qui :

Version du document
4 octobre 2021

- ▷ Déclare une variable "test" sur 32 bits sans lui attribuer de valeur
- ▷ Copie le contenu de l'adresse 0x00000080 de la mémoire dans cette variable

Exercice 4 **Comparaison**

Écrire un programme qui :

- ▷ Déclare deux variables, "A" et "B", dans lesquelles vous mettrez deux valeurs numériques de votre choix sur 32 bits.
- ▷ Déclare une variable "result", de 32 bits également
- ▷ Stocke 1 dans result si $A = B$, 0 sinon.

Exercice 5 **Somme des N premiers entiers**

Écrire un programme permettant de calculer dans une variable la somme des N premiers entiers. N sera lui aussi stocké dans une variable.

Exercice 6 **Les fonctions et le registre LR**

Analysez le code suivant :

```
SECTION INTVEC

B main

SECTION CODE

main
; Initialisons les registres à utiliser par FonctionAddition
MOV R0, #0xEE
MOV R1, #0x02
MOV R2, #0x0F

; Appelons notre fonction
BL FonctionAddition

; Plaçons le résultat dans le bon registre.
MOV R3, R0
B fin

; Fonction qui additionne trois nombres placés dans R0, R1, et R2
FonctionAddition
ADD R0, R0, R1
ADD R0, R0, R2
BX LR

fin
B fin

SECTION DATA
```

Que fait-il ?
Que font les instructions B, BL et BX ?

Exercice 7 Ecrire la fonction factorielle, qui calcule la factorielle d'un nombre placé dans R0 et qui retourne le résultat dans R1. Testez votre fonction.

Que se passe-t-il si je demande un nombre trop grand ?

Exercice 8 Ecrire une fonction permettant de remettre à zéro la mémoire à une adresse transmise via R0

Exercice 9 Ecrire une fonction permettant d'échanger les valeurs de deux variables dont les adresses sont transmises via R0 et R1.

3 Un peu plus difficile ...

Exercice 10 Calculer la somme des N premiers entiers pour N allant de 7 à 19.
Vous rendrez votre code "modulaire" en utilisant les exercices précédents.

Exercice 11 Ecrire un programme permettant de trouver le plus petit entier stocké dans un tableau.

Exercice 12 Ecrire un programme permettant de trier un tableau par ordre croissant.

4 Jouons un peu !

Exercice 13 Ecrire un programme permettant de vérifier une solution de sudoku.
Vous devrez vérifier :

- les lignes
- les colonnes
- les blocs