



TP01 - Introduction

Réponses aux questions

Quelle est la taille de chacune des variables ?

Il faut regarder dans le code lors de la déclaration de celles-ci.

Variable	incr	res	i
Tailles	4 Bytes	2 Bytes	4 Bytes

Quelle est la taille du code et comment procéder pour l'obtenir ?

Taille du code = 84 bytes Pour l'obtenir, il faut utiliser l'utilitaire "arm-none-eabi-size" et regarder le chiffre pour la section ".text".

Où se trouve chaque variable en mémoire (adresse absolue) ?

Pour connaître l'adresse exacte il faut utiliser : (gdb) p &<variable>

Variable	incr	res	i
Adresse	0x80003b04	0x80003b00	0x80003d00

Où se trouve le code en mémoire ?

Pour connaître l'adresse de la section ".text", il faut utiliser la commande : arm-none-eabi-size -format=sysv <file> Malheureusement pour une raison qui nous échappe, l'adresse est 0 pour chaque section. Comme nous pouvons le voir ci-dessous : main.o :

Section	size	addr
.text	84	0
.data	256	0
.bss	256	0
.ARM.attributes	20	0
Total	616	

Est-il possible d'améliorer l'algorithme ?

Pour optimiser le code, on peut sortir les instructions ldr et str de la boucle pour éviter d'accéder à la mémoire à chaque passage ce qui coûte en temps. On peut également éviter de mettre la valeur zéro dans l'espace mémoire i car comme i étant dans la section .bss sera de toute manière initialement à zéro.

Heures de travail

En dehors des heures de cours, nous avons dû travailler 3h. Durant le cours nous avons pu réalisé les tâches suivantes :

- Éditer le code à l'intérieur du fichier main.S
- Compiler le code, l'exécuter pas à pas, découvrir l'environnement
- Indiquer dans le code source la signification de chaque instruction
- Comprendre comment utiliser Git en ligne de commande

A la maison, nous avons dû finir les tâches suivantes :

- Optimiser le code sans changer l'algorithme
- Répondre aux questions
- Rédiger ce rapport



Synthèse

Non acquis

- Où se trouve le code en mémoire

Partiellement acquis

- Retenir ce que toutes les instructions font, la taille des variables

Acquis

- Comment fonctionne l'environnement : Git, Eclipse, Linux
- Ce que faisaient les différentes instructions
- La structure du code

Remarques & Feedback

Pour ce premier TP, il y avait beaucoup à apprendre et à se familiariser avec l'environnement et le code. Nous avons perdu un peu de temps pendant le cours pour chercher les réponses à nos questions. C'est un très bon premier travail pratique car nous avons appris beaucoup de choses et les instructions étaient claires.