

Projet informatique SICOM Livrable 2/4



SICOM-A
Le 4.11.2014

HONORE Antoine
PICHAT Maxence

Introduction

Pendant cette deuxième phase de travail, nous avons principalement travaillé à terminer le livrable précédent et à commencer à écrire la fonction *disasm*. Ce rapport contient un point sur l'avancement du travail et du groupe. Nous parlerons de notre organisation, de l'état du livrable et des principaux problèmes rencontrés.

I. Organisation

I.A. Mise en commun du travail

Pour le premier livrable nous avons fait le choix d'effectuer la mise en commun avec une dropbox. Ce système s'est très vite avéré compliqué à gérer. Nous n'avions en effet aucun moyen de travailler en même temps sur le projet. Si l'un d'entre nous écrit une fonction l'autre ne peut plus compiler car la fonction de l'autre n'est pas syntaxiquement correcte. Nous téléchargeons donc la dropbox sur nos ordinateurs personnels avant de partager le code valide sur la dropbox. Ce système est le même que le logiciel **git**. C'est la solution que nous avons choisie, nous reviendrons sur **git** dans la partie « problèmes rencontrés ».

I.B. Répartition des rôles

Le livrable précédent n'était pas terminé. Maxence s'est occupé d'écrire les fonctions *set* et *assert* et Antoine a travaillé sur la fonction *disasm*. Les fonctions *set* et *assert* ayant été écrites suffisamment tôt, Maxence a eu le temps d'écrire le plus gros du dictionnaire.

II. Le livrable

II.A. Etat du livrable

Ce livrable est plutôt bien avancé. Nous avons eu le temps de modifier la commande *disp* pour lui faire afficher la mémoire octets par octets. La commande *disasm* fonctionne pour les instructions ADD, ADDI et NOP. Les autres fonctions ont été rentrées dans le dictionnaire mais nous n'avons pas écrit de code assembleur pour les tester.

II.B. Problèmes rencontrés

II.B.i. Dictionnaire

En ce qui concerne le dictionnaire et sa lecture, nous avons fait face au problème de classer les fonctions en types R, I et J et bien décrire leurs opérandes pour que la fonction qui lise le dictionnaire reconnaisse bien toutes les caractéristiques de chaque fonction.

II.B.ii. Git

Comme nous en parlions dans la description du mode de partage des fichiers, l'utilisation de git a soulevé des problèmes, et il nous a fallu du temps pour nous adapter et faire marcher la synchronisation. En effet, il faut soumettre les changements que l'on a fait avant de télécharger ceux faits par l'autre, ce qui s'est révélé compliqué : des conflits « merge » entre des fichiers que nous avons tous les deux modifiés apparaissaient et ne permettaient pas la synchronisation. Il faut donc veiller à bien télécharger les dernières modifications de son partenaire avant de commencer à retravailler les fichiers.

Il faut modifier le fichier `./git/config` de telle sorte que

url = <https://github.com/username/depot> devienne

url = `https://username@github.com/username/depot`

Sinon git ne nous laisse pas accéder au dépôt en ligne pour « pusher » nos changements.

II.B.iii. Code

Pour ce qui est de la fonction `disasm`, les utilisations de `strtok` se sont révélés être piégeuses, puisque nous ne savions pas qu'elles modifient la chaîne de caractère au fur et à mesure qu'on avance dans la ligne.

Nous avons eu un peu de mal à nous familiariser avec la notion d'*union* mais nous avons trouvé les solutions assez vite.

II.B.iv. Tests

Nous nous sommes lancés dans l'utilisation des tests automatisés pour valider le fonctionnement des programmes. Ceci est une phase du projet assez fastidieuse puisqu'il faut trouver les cas les plus tordus pour voir la réaction du programme.