Troisième livrable – Projet Info SEI

Simulateur MIPS

Nicolas SCHOEMAEKER
Eric FELTRIN

Grenoble INP – Phelma
2A - SEI

Introduction

A l'issu du second livrable nous avons mis en place la commande lp permettant le chargement d'un fichier en mémoire et de pouvoir modifier cette dernière grâce à la commande lm ainsi qu'une commande da, permettant de retrouver le code assembleur d'une ligne d'instruction en code hexadécimal.

L'objectif de ce troisième livrable est l'exécution du programme assembleur en utilisant la commande si permettant d'exécuter une ligne d'instruction ou s identique à la fonction si sauf s'il s'agit d'une procédure qui sont elles exécutées complètement, enfin la commande run permettant d'exécuter l'ensemble des instructions avant le prochain break point. Dans ce livrable les commandes relatives aux break points (bp création d'un break point, er suppression d'un break point et db affichage des break points) ont également été implémentées.

Implémentation de l'incrément

Exécution du code assembleur

Une fonction permettant de choisir la fonction à utiliser selon l'instruction à exécuter (int execute(mips *)) est utilisé pour les trois commandes d'exécution : run, s, si. Pour choisir la fonction relative à l'instruction assembleur à exécuter nous nous sommes servis comme pour implémentation de da, des code d'opération op_code de chaque instructions. La commande run boucle jusqu'à atteindre un break point ou la fin du code et la commande s en cas de procédure boucle jusqu'à revenir à l'adresse suivant la procédure.

Les break points

Pour mémoriser les break points nous utilisons une liste triée par ordre croissant d'adresse permettant de facilement connaître le prochain break point en comparant la valeur courante du PC aux différentes valeurs de la liste.

Instructions assembleurs

Pour chaque instruction, un fichier a été crée afin de faciliter l'ajout d'une nouvelle instruction (juste ajouter dans le fichier execute.h le include et la fonction avec son op_code. Dans ces fichiers se trouvent donc une unique fonction effectuant les opérations sur les registres et la mémoire du mips : c'est le seul endroit où l'on modifie le mips.

Teste du programme

Nous avons implémenté un test de la commande run : appelé run.simcmd exécutant le fichier exempleELF.o (simple execution). Ensuite boucle.simcmd (bouclage avec l'instruction BNE) et procedure.simcmd (execution d'une procédure).

Conclusion

Nous sommes désormais capable de d'exécuter des fichiers .o préalablement chargés (livrable 2). Ce qui nous rapproche du but en effet jusqu'à cet incrément tout ne travail n'avait pas de réelle application puisque on ne pouvais voir notre simulateur en fonctionnement c'est chose faite avec cet incrément. Cependant sans les incrément précédent il est impossible d’espérer avoir le moindre résultat ce qui prouve encore une fois leur utilité.