

ARCHITECTURE ET SYSTÈME: TP 3 (NOTÉ)

Licence Info 2

Flavien Breuvar, Stefano Guerrini, Antoine Kaszczyk et Lê Thành Dũng Nguyễn

A.A. 2018–19 - 23/10/2018

Communiquer la composition de votre binôme en début de TP (marquer nom, prénom, date de naissance et numéro étudiant des membres du binôme sur la feuille fournie par l'enseignant).

Écrire la solution de chaque exercice dans un fichiers `exN.asm`, où N est le numéro de l'exercice. Le fichier devra contenir le code assembleur de la solution (dûment commentée), être compilable par l'assembleur et correspondre à un petit programme qui pourra être exécuté par le simulateur de LC-3. En tête au tout fichier marquer dans un commentaire nom, prénom, date de naissance et numéro étudiant

des deux membres du binôme.

Envoyer les fichiers en utilisant l'interface pour l'envoi des devoirs de votre groupe de TP (sur la page du cours de l'ENT).

La remise des devoirs termine à 12h00.

NB. En cas de devoirs similaires ou de suspicion de triche, les étudiants seront convoqués par les enseignants et des sanctions seront appliquées.

Exercice 1. Écrire dans un fichier le code assembleur qui exécute dans l'ordre les actions suivantes:

1. mettre à zéro le registre R5;
2. copier la valeur du registre R0 dans le registre R2;
3. copier la valeur contenue dans la location de mémoire x vers la location de mémoire y (utiliser le registre R1 comme registre d'appui pour cette opération);
4. copier dans la location z l'adresse de la location x . (*Attention:* pour le faire, il faut d'abord copier l'adresse dans un registre interne de l'UE. Vous pouvez utiliser le registre R5.)

Le code devra contenir les directives assembleurs qui réservent de la place pour x , y et z .

Exercice 2. Écrire un programme assembleur LC-3 qui met dans la location de mémoire y la valeur absolue de l'entier contenu dans la location x . (*Suggestion:* LC-3 ne permet pas des opération arithmétique ou logique directement sur les valeurs dans une location de mémoire, donc il faudra d'abord transférer la valeur de y dans un registre, calculer la valeur absolue et finalement recopier la valeur dans la location du résultat.) Le code devra contenir les directives assembleurs qui réservent de la place pour x et y .

Exercice 3. Écrire un programme assembleur LC-3 qui inverse un tableau mémorisé à l'adresse `tab1` dans un nouveau tableau à l'adresse `tab2` (le premier élément de `tab1` devient le dernier de `tab2`, le deuxième devient l'avant dernier, etc.). La taille des tableaux est mémorisée à l'adresse `taille`. Le code devra contenir les directives assembleurs qui réservent de la place pour mémoriser la taille des tableaux et les deux tableaux.

Exercice 4. Mettre la solution de l'exercice 3 sous la forme d'une sous-routine `inverse`. La sous-routine reçoit la taille des tableaux dans le registre R0 et les adresses des deux tableaux dans les registres R1 et R2. La sous-routine peut modifier (donc pas besoin de les sauvegarder) tous les registres dont a besoin pour calculer l'inverse. Le programme principale devra tout simplement exécuter la sous-routine pour copier (comme dans l'exercice 3) le tableau mémorisé à l'adresse `tab1`, dont la taille est mémorisée à l'adresse `taille`, dans un nouveau tableau à l'adresse `tab2`. Compléter le programme avec les directives qui réservent de la place pour mémoriser la taille des tableaux et les deux tableaux.

Exercice 5 (Bonus). Reprendre la solution de l'exercice 4. En utilisant la pile pour sauvegarder et récupérer les valeurs des registres touchés par la routine, assurer que après l'exécution de la sous-routine tous les registres ont repris les valeurs qu'ils avaient au moment de l'appel de la sous-routine. Compléter le programme avec les directives qui réservent de la place pour la pile et le code pour son initialisation.