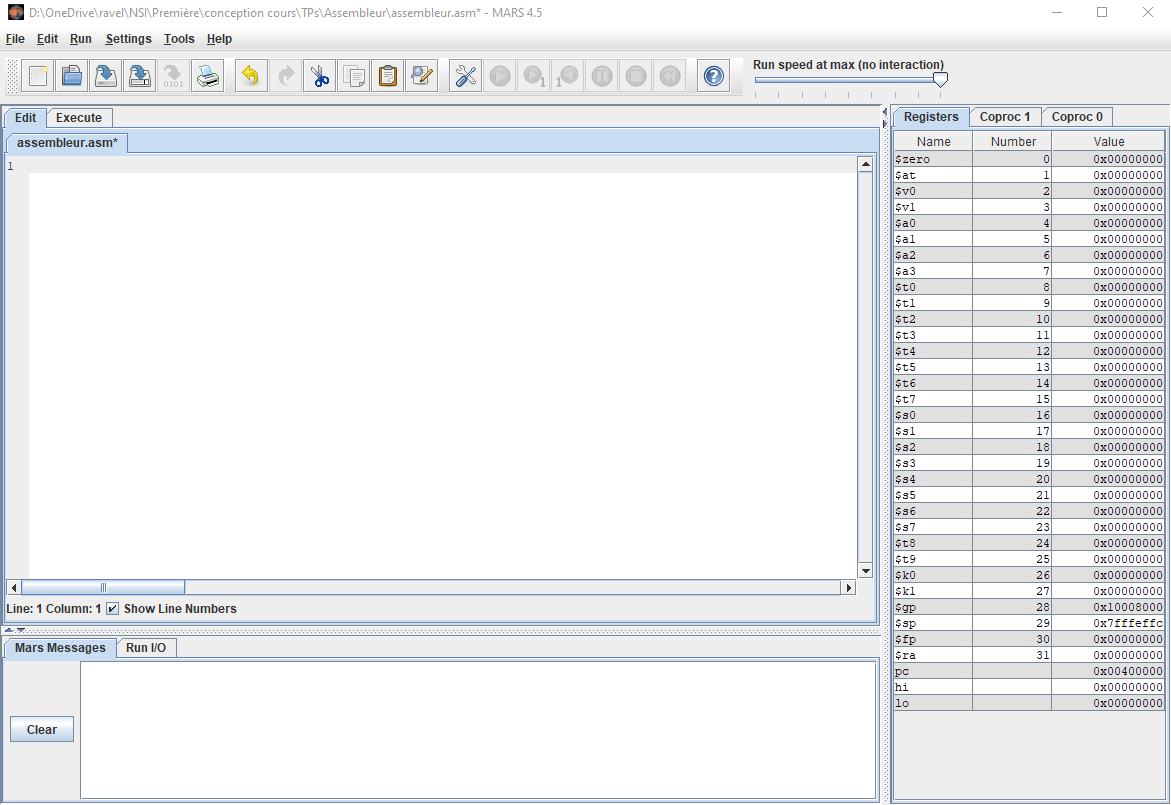
Mode d’emploi de MARS

# Interface



Zone d’écriture des messages et résultats écrits

**Zone des registres**

Zone d’écriture du programme

**1 2 3 4 5 6 7**

1 : Nouveau fichier

2 : Ouvrir un fichier

3 : Enregistrer

4 : Enregistrer sous

5 : Assembler

6 : Exécuter l’ensemble du programme

7 : Exécuter une ligne du programme

# Ecrire un premier programme

Un programme Assembleur contient potentiellement plusieurs sections. Nous n’utiliserons que deux de celles-ci (.data pour déclarer des données, et .text pour indiquer le début du programme suite à l’utilisation de données). Dans MARS les registres $zero, $at, $v0, $v1, $ai, $k0, $k1, $gp, $sp, $ra sont réservés à des usages spécifiques. Exemple de programme qui effectue 2+3 et stocke le résultat dans le registre $t0.

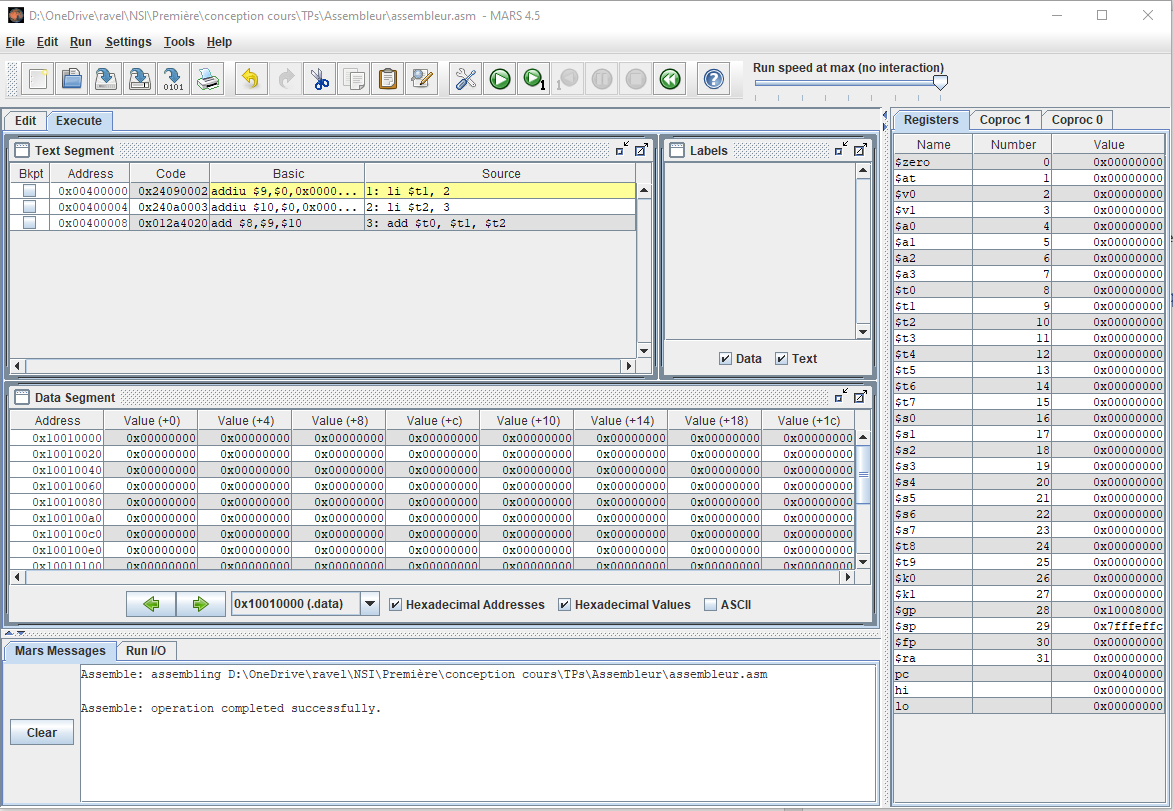
li $t1, 2

li $t2, 3

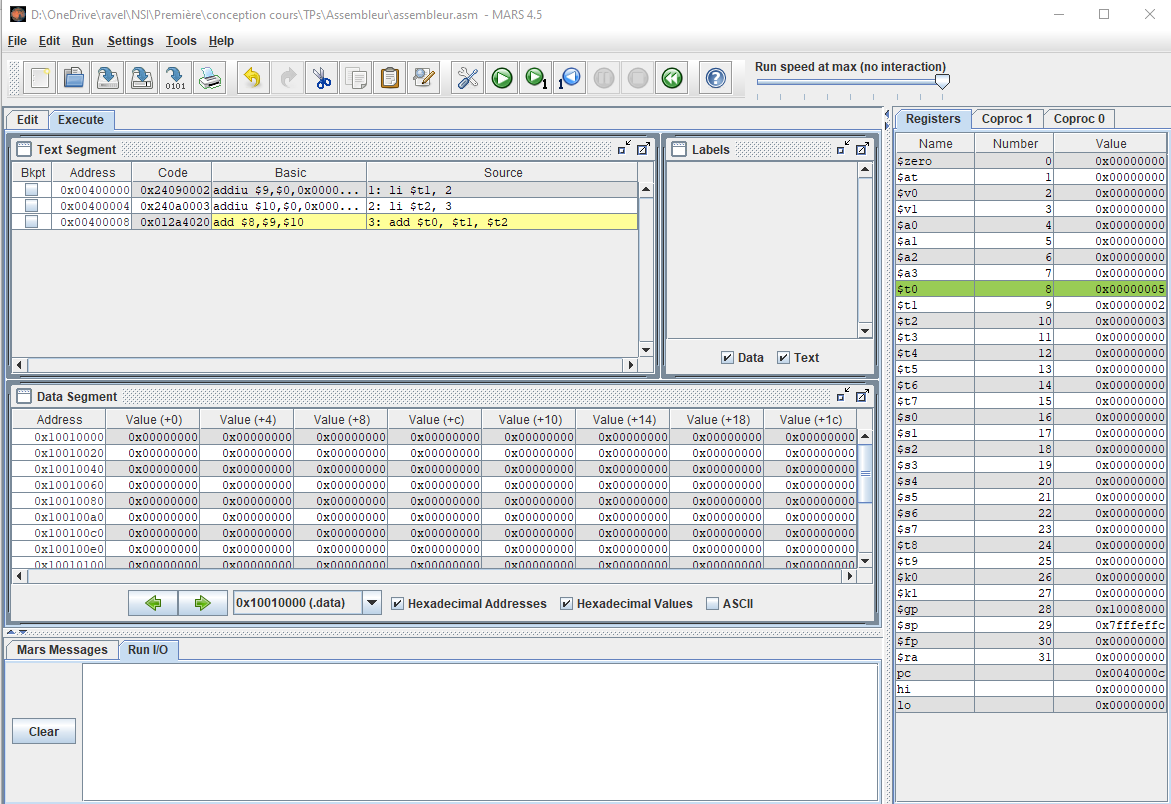
add $t0, $t1, $t2

# Assembler et exécuter

Une fois tapé, il faut enregistrer le programme et l’assembler. On obtient alors une deuxième fenêtre qui est la fenêtre d’exécution.



En cliquant sur l’icône « Exécuter une ligne du programme » on voit le programme s’exécuter pas à pas. Et on observe les modifications des contenus de chaque registre.



# Quelques commandes fréquentes

|  |  |
| --- | --- |
| Commande | Signification |
| li | Charger un nombre dans un registre |
| la | Charger l’adresse d’une variable |
| lb, lw | Charger le byte ou le word se trouvant à l’adresse en référence |
| add | Additionner |
| sub | Soustraire |
| beq | Aller si égal |
| bne | Aller si différent |
| li $v0, 1  add $a0, $t0, 0  syscall | Affiche le contenu de $t0 dans la console |
| j | Aller à |

Pour déclarer un « label » (un point de repère dans le programme), on met son nom suivi de « : ». Pour y aller on met juste son nom. Exemple :

debut :

…

Suite d’instruction

…

j debut