
Travaux Pratiques 1 : Programmation en mini-assembleur

Il est demandé, pendant ce TP, d'écrire des programmes simples en mini-assembleur et de les exécuter sur un processeur simulé par le logiciel `amil` (*assembleur miniature pour l'informatique de licence*).

1 Prise en main : le terminal

Dans un terminal, taper les lignes de commandes suivantes :

<code>~boudes/pub/EI/init.sh</code>	Exécuter un script qui modifiera légèrement votre environnement de travail pour l'adapter au TP.
<code>mkdir TP1</code>	Créer un répertoire TP1.
<code>cd TP1</code>	Entrer dans le répertoire TP1.
<code>amil &</code>	Lancer <code>amil</code> , le simulateur de mini-assembleur en <i>tâche de fond</i> (en tâche de fond le terminal ne restera pas bloqué jusqu'à l'arrêt du simulateur).

Astuces du terminal. la touche de tabulation vous permet de compléter votre saisie quand vous tapez une commande dans le terminal. La touche flèche vers le haut rappelle une ligne de commande tapée précédemment.

2 Initialisation de la mémoire

Soit la case mémoire, x , d'adresse 10 et la case mémoire, y , d'adresse 11. Écrire et exécuter le programme qui initialise x à 7×2 et y à $x - 1$.

3 Exécution conditionnelle d'instructions

À l'aide de l'instruction `sautpos`, écrire les programmes correspondant aux algorithmes suivants et les exécuter avec `amil` sur un exemple, afin de tester leur correction :

1. Soient la valeur a à l'adresse 20, b à l'adresse 21. Si $a < b$ alors écrire a à l'adresse 22 sinon écrire b à l'adresse 3.
2. Soient trois cases mémoires contenant trois entiers. Calculer et écrire le minimum de ces trois entiers en mémoire.

4 Boucles d'instructions

1. Avec l'instruction `saut`, écrire un programme qui ne termine jamais.
2. Avec l'instruction `sautpos`, écrire un programme qui ne termine jamais.

On dit de ces programmes qu'ils bouclent à l'infini.