

# Architecture des ordinateurs - TP2

- Baptiste RODRIGUES
- Etienne CHAZELLE

## Introduction

Ce compte-rendu contient les réponses aux questions du deuxième TP d'architecture des ordinateurs. Les exercices obligatoires sont précisés en italique. L'objectif de ce TP était de découvrir les bases de la programmation en assembleur. Dans ce sens, nous avons d'abord répondu à des questions sur les registres, sur les instructions, et enfin sur la mémoire. Nous avons ensuite analysé notre premier programme en assembleur, avant de le modifier pour mieux comprendre son fonctionnement. Ensuite, nous avons réalisé notre premier programme en assembleur, qui a pour but de sortir deux chaînes de caractères différentes sur la sortie standard.

### Exercice 3

1. `mov bh, 0xAB`
2. `mov bl, 0xCD`

Avec ces deux instruction le registre `ebx` est égal à **0x0000ABCD**.

### Exercice 4

Instruction	Registre
<code>mov eax, 0xf00000f0</code>	<code>ax = 0x00f0</code>
<code>add ax, 0xf000</code>	<code>eax = 0xf000f0f0, ax = 0xf0f0, ah = 0xf0, al = 0xf0</code>
<code>add ah, al</code>	<code>eax = 0xf000e0f0, ax = 0xe0f0, ah = 0xe0, al = 0xf0</code>

Après l'exécution des instructions, les valeurs des registres sont les suivants :

registre	valeur
<code>eax</code>	<code>0xf000e0f0</code>
<code>ax</code>	<code>0xe0f0</code>
<code>ah</code>	<code>0xe0</code>
<code>al</code>	<code>0xf0</code>

### Exercice 6

L'instruction `mov eax, 3` déplace la valeur 3 dans le registre **`eax`**, tandis que `mov eax, [3]` déplace la valeur de l'adresse 3 en mémoire (0x000000003) dans le registre **`eax`**.

### Exercice 8

Pour chaque étapes voici la valeur du registre eax :

Instruction	Registre
mov eax, 5	0x00000005
sub eax, 1	0x00000004
mov al, [eax]	0x00000010
mov ah, [eax]	0x00007f10

Après ces instructions le registre eax possède comme valeur 0x00007f10.

Exercice 10

1.  
[0] = 0x01 [1] = 0x00 [2] = 0x02 [3] = 0x00
2.  
[0] = 0x01 [1] = 0x21 [2] = 0x02 [3] = 0x00
3.  
[0] = 0x01 [1] = 0x21 [2] = 0x01 [3] = 0x00

Exercice 12 & 14

Voir les programmes dans le ZIP.

Conclusion

Pour conclure, nous avons rencontré plusieurs difficultés à différentes étapes de ce TP. Dès le début, des problèmes sont apparus lors des conversions, puis avec la gestion des instructions et des registres, notamment lors de l'exercice 8. En fin de TP, la création des programmes s'est avérée complexe, particulièrement durant l'exercice 12, où le calcul du nombre de caractères à afficher posait problème. Un point bloquant a été le retour à la ligne, qui ne fonctionnait pas correctement, jusqu'à ce que nous identifions l'erreur.