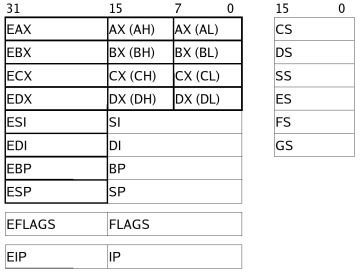
**LM semaine 3**

*Les Registres divisible*

**1 octets = 8 bits**

EAX est divisible en AX,AH et AL

* AX = les 16 premiers bits (2 octets)
* AL = les 8 premiers bits (1 octets)
* AH = les 8 suivants (du 7 au 15) (1 octets)

ABX est divisible en BX, BH et BL

ECX est divisible en CX, CH et CL

EDX est divisible en DX, DH et DL

*Le code ASCII*

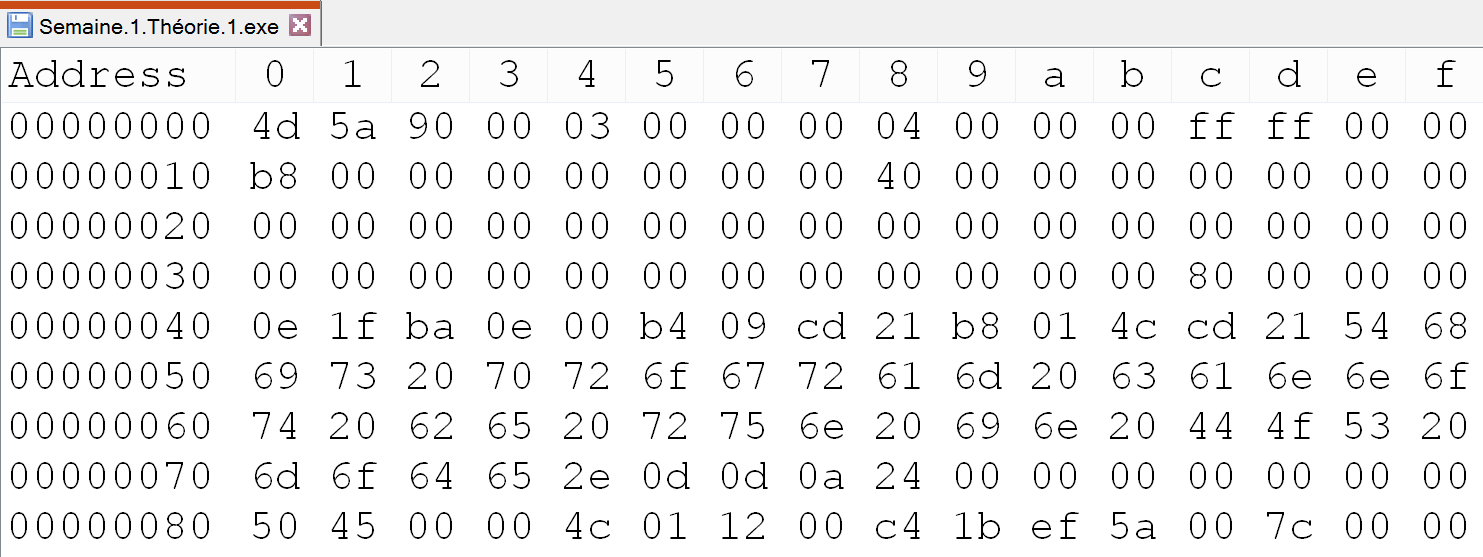
Chaque caractère est codé sur 1 octet et chaque caractère a un numéro hexadécimal

*(pour coder n caractères il faut n octets(si on veut coder ‘hello’ il faudra h+E+L+L+O donc 5 octets car 5 caractères)*

*Le tableau n’est pas à apprendre par cœur*

L’adressage absolu

Chaque octet en mémoire se trouve à une adresse, pour accéder à un octet il faut donc connaitre son adresse



Pour trouver l’adresse il suffit d’additionner la colonne adresse de gauche avec la ligne adresse du haut

Ex : 0xff se trouve a l’adresse 00000000 et c donc 000000c

*les directives dans la section data*

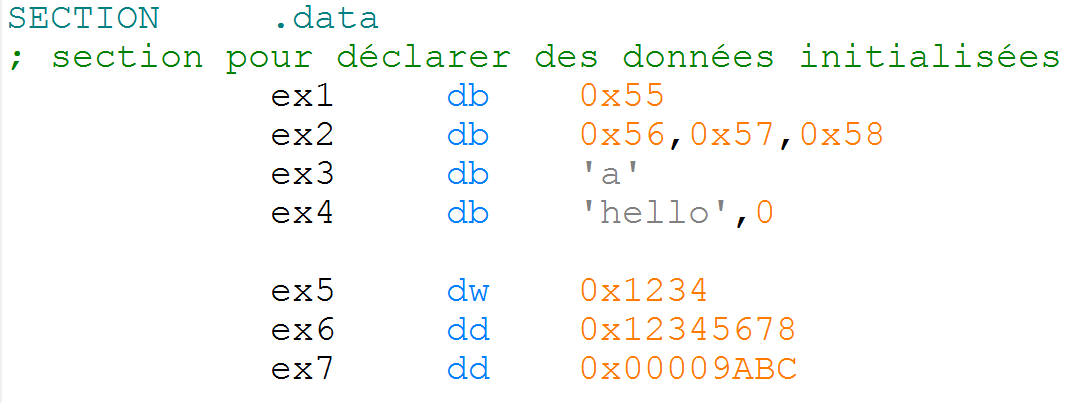
db= define byte (8 bits) = byte

dw=define word (16 bits)= word

dd= define double word (32 bits) = dword

la syntaxe dans la section .data sera donc

variable [db|dw|dd] valeur

**ex :**

**Pour mettre une nouvelle valeur en mémoire on utilise :**

Mov dword [ex1],0x77

Dans la mémoire on copiera donc 0x77 a l’emplacement de ex1

Les [] sont obligatoire pour copier en mémoire ainsi que les directives afin de savoir le nombre de bits à prendre dans ca la c’est 32 bites donc 4 octets(car dd) la valeur commencer donc a ex1 et se terminera lorsque cela fera 4 octets

mov dword [ex7],0x77

* + Copie la valeur 7716 en mémoire sur 4 octets à partir de l’adresse « ex7 »

mov word [ex5],0x55

* + Copie la valeur 5516 en mémoire sur 2 octets à partir de l’adresse « ex5 »

**La différence entre 2 instructions**

* **mov ebx,ex1** Copie l’adresse « ex1 » dans le registre ebx sur 4 octets (32 bits)
* **mov byte [ebx],0x33**

Copie la valeur 3316 en mémoire sur 1 octet à l’adresse donnée par la valeur du registre ebx

Voc

Bus d’adresse : l’endroit ou on trouvera l’emplacement de la valeur

Bus de données : l’endroit ou on trouvera la valeur

RAM : RAM (Random Access Memory)

Le processeur (CPU) : contient les registres

Bus de contrôle : permet de contrôler le type de l’opération sur le bus (ex read or write)