

10 juillet 2016

1. DB: directive qui permet l'allocation en mémoire sur 1 octet (byte) ou une suite d'octets.

DW: ... sur 2 octets ou une suite de mots

DD: ... sur 4 octets ou ... et double.

telle DB "bienvenue"; définit un tableau dont la valeur de chaque case n'est autre que le code ASCII sur 1 byte de chaque lettre

trois DW 4,5,6; définit un tableau de 3 mots initialisé par les valeurs 4,5 et 6.

nombre DD 12345678h; alloue un espace de 4 bytes pour stocker la valeur 12345678h

2. 1<sup>o</sup>) bienvenue, ceci est nouveau

2<sup>o</sup>) ceci est nouveau

3<sup>o</sup>) nouveau

3. 5 étapes pour écrire et exécuter un programme en assembleur

- Saisie du code source à l'aide d'un éditeur de texte sous DOS
- Sauvegarde du fichier avant de sortir de l'environnement
- Compilation du programme
- Edition des liens pour avoir un programme exécutable
- Exécution du programme

4. ~~assume~~ `cs: code`, `ds: data`, `ss: pile`

la directive `assume` permet d'indiquer à l'assembleur quel est le segment de données et celui de code afin qu'il génère des adresses correctes.

5. Différence entre "Procédure" et "Macro".

- Un macro est un ensemble d'instructions qui se répètent dans un programme et qui peut être référencé au milieu du programme.
- Une procédure est un bloc d'instructions repéré par un nom et qui est appelé par `CALL` pendant l'exécution d'un programme.

6. Instructions `CALL` et `RET` dans une procédure à quoi elles servent et action ?

`Call` : permet de faire appel à une procédure.

`Push` Place la valeur du registre `IP` dans la pile



et (qui) charge l'adresse de la 1<sup>ère</sup> instruction de la procédure dans IP

Ret: Permet le retour au programme principal.  
depile la pile et réinitialise la valeur d'IP  
(6to)

7.  $AX = (5BX + CX/3) / DX + 4$

pour faire

5BX  
↓  
BX

MOV AX, BX.

MUL 5

MOV BX, AX

$\frac{CX}{3}$

→ CX

MOV AX, CX

DIV 3

MOV CX, AX

ADD BX, CX ;

MOV AX, BX

DIV DX

ADD AX, 4

8. PUSH et POP ?

PUSH : instructions permettant d'empiler

POP : instructions permettant de dépiler

## IN et OUT:

Ces instructions permettent de lire et d'écrire sur un port d'E/S.

### 9. Soustraire $N_1$ et $N_2$ .

mov AX, [3000H]

SUB AX, [3100H]

JS negahf

JZ nul

MOV [3200H], AX

JMP fin

negahf : mov [3300H], AX

JMP fin

nul : mov [3400H], AX

fin : hlt

### 10. ~~La~~ Somme de 100, 200, 300 dans AX

#### Programme principal

MOV AX, 100

push AX

MOV AX, 200

push AX

Call Somme

Somme PROC

push bp

MOV sp, bp

MOV AX, [bp+4]

ADD AX, [bp+6]

MOV AX, [bp+8]

pop bp

~~pop bp~~

ret 6

Somme ENDP