

Projet TP ARCHI2

Réalisé par :

Hiouani Lydia 202031060760 G3 Lydiahiovani@gmail.com

Lamri Mohamed Said Aimen 202031025221 G4

lamrimohamedsaidaimen@gmail.com

Exercice01 :

1. Charger en mémoire la chaîne de caractère suivante à partir de l'adresse 12Ah :1E00h

'Le module architecture des ordinateurs est un module fondamental de la spécialité informatique '

en utilisant la commande e (enter) on introduit le code ASCII des caractères de la chaîne dans la mémoire à partir de l'adresse 012A :1E00

-e 012A:1E00									
012A:1E00	EB.4C	02.65	B4.20	8F.6D	88.6F	26.64	B1.75	56.6C	
012A:1E08	A2.65	A8.20	56.61	FE.72	06.63	41.68	57.69	C6.74	
012A:1E10	06.65	A6.63	56.74	02.75	E8.72	B6.65	05.20	E8.64	
012A:1E18	56.65	05.73	8A.20	45.6F	02.72	3C.64	C0.69	72.6E	
012A:1E20	44.61	C6.74	05.65	01.75	80.72	3E.73	41.20	57.65	
012A:1E28	02.73	75.74	0F.20	24.75	18.6E	0C.20	06.6D	80.6F	
012A:1E30	3E.64	A8.75	56.6C	00.65	75.20	11.66	0C.6F	01.6E	
012A:1E38	EB.64	0D.61	24.6D	07.65	0C.6E	50.74	80.61	3E.6C	
012A:1E40	A8.20	56.64	00.65	75.20	02.6C	0C.61	58.20	88.73	
012A:1E48	45.70	01.65	E9.63	05.69	05.61	E8.6C	32.69	ED.74	
012A:1E50	B9.65	04.20	00.69	E8.6E	55.66	F0.6F	72.72	0D.6D	
012A:1E58	FE.61	0E.74	B1.69	56.71	80.75	06.65	B0.20		

-Vérifiez que votre commande a été correctement exécutée.

En utilisant la commande d(dump) on visualise le contenu de la mémoire et on vérifie que la commande précédente a été exécutée correctement

-d 012A:1E00 1E60																
012A:1E00	4C	65	20	6D	6F	64	75	6C-65	20	61	72	63	68	69	74	Le module archit
012A:1E10	65	63	74	75	72	65	20	64-65	73	20	6F	72	64	69	6E	ecture des ordin
012A:1E20	61	74	65	75	72	73	20	65-73	74	20	75	6E	20	6D	6F	ateurs est un mo
012A:1E30	64	75	6C	65	20	66	6F	6E-64	61	6D	65	6E	74	61	6C	dule fondamental
012A:1E40	20	64	65	20	6C	61	20	73-70	65	63	69	61	6C	69	74	de la specialit
012A:1E50	65	20	69	6E	66	6F	72	6D-61	74	69	71	75	65	20	56	e informatique U
012A:1E60	02															.

2. Avec une séquence d'instructions optimale stockée à partir de l'adresse 12Ah: 4F00h , remplacez les caractères 'n' et 'N' par un trèfle et le 'm' et 'M' par un pique

```

-A 012A:4F00
012A:4F00 mov bx,1DFF
012A:4F03 mov cx,60
012A:4F06 inc bx
012A:4F07 cmp byte ptr [bx],6E
012A:4F0A jnz 4F12
012A:4F0C mov byte ptr [bx],05
012A:4F0F dec bx
012A:4F10 jmp 4F06
012A:4F12 cmp byte ptr [bx],6D
012A:4F15 jnz 4F1D
012A:4F17 mov byte ptr [bx],06
012A:4F1A dec bx
012A:4F1B jmp 4F06
012A:4F1D loop 4F06
012A:4F1F

```

- Vérifiez que le programme s'est bien exécutée.

On utilise la commande d

```

-G= 012A:4F00 4F1F

AX=0000 BX=1E5F CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=012A ES=073F SS=073F CS=012A IP=4F1F  NU UP EI NG NZ AC PO CY
012A:4F1F FC          CLD
-D 012A:1E00
012A:1E00 4C 65 20 06 6F 64 75 6C-65 20 61 72 63 68 69 74  Le .odule archit
012A:1E10 65 63 74 75 72 65 20 64-65 73 20 6F 72 64 69 05  ecture des ordi.
012A:1E20 61 74 65 75 72 73 20 65-73 74 20 75 05 20 06 6F  ateurs est u. .o
012A:1E30 64 75 6C 65 20 66 6F 05-64 61 06 65 05 74 61 6C  dule fo.da.e.tal
012A:1E40 20 64 65 20 6C 61 20 73-70 65 63 69 61 6C 69 74  de la specialit
012A:1E50 65 20 69 05 66 6F 72 06-61 74 69 71 75 20 B0 56  e i.for.atiqu .U
012A:1E60 02 89 16 B2 56 E9 EA 04-E8 17 ED B9 02 00 E8 3A  ....U.....:
012A:1E70 F0 72 1B 8A C2 3C 03 74-EC FE 06 B1 56 E9 21 01  .r...<.t....U.!.

```

Remarque :

Dosbox n'affiche pas les caractères spéciaux et donc on remarque des points et pas des trèfles et piques à la place de 'm' et 'n'

3. Visualiser le code de la séquence d'instructions écrite dans la question 2

```

-u 4F00 4F1F
012A:4F00 BBFF1D      MOV     BX,1DFF
012A:4F03 B96000      MOV     CX,0060
012A:4F06 43          INC     BX
012A:4F07 803F6E      CMP     BYTE PTR [BX],6E
012A:4F0A 7506      JNZ     4F12
012A:4F0C C60705      MOV     BYTE PTR [BX],05
012A:4F0F 4B          DEC     BX
012A:4F10 EBF4      JMP     4F06
012A:4F12 803F6D      CMP     BYTE PTR [BX],6D
012A:4F15 7506      JNZ     4F1D
012A:4F17 C60706      MOV     BYTE PTR [BX],06
012A:4F1A 4B          DEC     BX
012A:4F1B EBE9      JMP     4F06
012A:4F1D E2E7      LOOP    4F06
012A:4F1F FC          CLD

```

4. Déduire la taille de la mémoire occupée par votre séquence d'instructions.

Taille_mem_occup=@fin-@debut+1=20 octets

Exercice02 :

1. Écrire en mémoire la séquence d'instructions

```

-A 1EFE:0100
1EFE:0100 MOV DX,0D
1EFE:0103 MOV BX,1111
1EFE:0106 MOV WORD PTR [BX],BX
1EFE:0108 MOV CL,3
1EFE:010A SHL WORD PTR [BX],CL
1EFE:010C DEC DX
1EFE:010D ADD BX,2
1EFE:0110 MOV CX,DX
1EFE:0112 LOOP 0106

```

2. Que fait cette séquence ?

Cette séquence fait des multiplications de bx par 8 et incrémente bx de 2

3. Visualisez le contenu de la structure (juste la structure et pas une case

mémoire de plus) avant l'exécution de la séquence

```

-D 458F:1111 1128
458F:1110 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
458F:1120 00 00 00 00 00 00 00 00-00 .....

```

4. Même question après exécution de la séquence tout en commentant le résultat obtenu

```

-G=1EFE:0100 0114
AX=0000 BX=1129 CX=0000 DX=0001 SP=00FD BP=0000 SI=0000 DI=0000
DS=458F ES=073F SS=073F CS=1EFE IP=0114 NU UP EI PL NZ NA PO NC
1EFE:0114 0000 ADD [BX+SI],AL DS:1129=00
-D 458F:1111 1128
458F:1110 88 88 98 88 A8 88 B8-88 C8 88 D8 88 E8 88 F8 .....
458F:1120 88 08 89 18 89 28 89 38-89 .....(8.

```

Le résultat obtenu : 88 88 98 88 A8 88 B8-88 C8 88 D8 88 E8 88 F8

5. Donnez l'état du processeur après la première itération de la boucle tout en commentant les bits d'état (flags)

```

AX=0000 BX=1113 CX=000C DX=000C SP=00FD BP=0000 SI=0000 DI=0000
DS=458F ES=073F SS=073F CS=1EFE IP=0112 NU UP EI PL NZ NA PO NC
1EFE:0112 E2F2 LOOP 0106

```

L'état des flags :

OF	0
DF	0
IF	1
SF	0
ZF	0
AF	0
PF	0
CF	0

Exercice03 :

1. Générez la séquence d'instructions correspondant au code machine.

Génération de la séquence d'instructions suivantes :

A1 00 00 03 06 02 00 A3 04 00 72 09 BA 19 00 B4 09 CD 21 EB 07 BA 06 00 B4 09 CD 21 B4 4C CD 21

En utilisant la commande E on fait rentrer la séquence d'instructions précédentes octet par octet (chaque octet est représenté par deux numéros en hexadécimal) on tape à chaque fois espace pour faire entrer l'octet suivant comme suit :

```
-E 0000
073F:0000 A1.    00.    00.    03.    06.    02.    00.    A3.
073F:0008 04.    00.    72.    09.    BA.    19.    00.    B4.
073F:0010 09.    CD.    21.    EB.    07.    A3.BA 06.    00.
073F:0018 B4.    09.    CD.    21.    B4.    4C.    CD.    21.
```

Pour générer la séquence d'instruction on utilise la commande U suivit de l'adresse où est stocker la première instruction comme suit :

```
-U 0000
073F:0000 A10000      MOV     AX,[0000]
073F:0003 03060200    ADD     AX,[0002]
073F:0007 A30400      MOV     [0004],AX
073F:000A 7209       JB      0015
073F:000C BA1900      MOV     DX,0019
073F:000F B409       MOV     AH,09
073F:0011 CD21       INT     21
073F:0013 EB07       JMP     001C
073F:0015 BA0600      MOV     DX,0006
073F:0018 B409       MOV     AH,09
073F:001A CD21       INT     21
073F:001C B44C       MOV     AH,4C
073F:001E CD21       INT     21
```

2. Avant le lancement de l'exécution de la séquence il faut que :

a. les registres suivants sont initialisés comme suit : DS=2F1h, IP=56EH, CS=5221H :

On initialise les registre DS,IP,CS avec la commande R + « le nom du registre », (R DS, R IP, R CS pour les trois registre) comme suit :

```
-rds
DS 073F
:2F1
-rs
CS 073F
:5221
-rip
IP 0100
:56E
```

B. Les emplacements mémoire référencés en lecture dans la séquence d'instructions contiennent les valeurs suivantes F214H,0122h

On doit stocker ces valeur dans l'emplacement mémoire qui ont l'adresse 0000 ,0002 car les deux premières instructions du programme font un adressage direct dans le segment de donne vers les deux adresses 0000 et 0002

On utilise la commande E suivit de l'offset 0000 car l'adresse segment est déjà initialise a 02F1H

Comme suit :

```
-e 0000
02F1:0000 14. F2. 3D.22 75.01
```

3. Vérifiez que vos commandes de la question 2 sont bien exécutées.

Pour vérifier que les registres ont bien été modifier on utilise la commande r comme suit :

```
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=02F1 ES=073F SS=073F CS=5221 IP=056E NU UP EI PL NZ NA PO NC
5221:056E 0000 ADD [BX+SI],AL DS:0000=80
```

On peut voir clairement que tous les registres sont modifier.

Pour vérifier que le contenu des deux emplacements mémoires ont bien été modifier on utilise la commande d suivit de l'adresse segment de donner et l'offset du début ainsi que la fin puisque on fait entrée 4 octet après l'offset du début 0000 donc l'offset de la fin est

4 me suit :

```
-D 2F1:0000 0004
02F1:0000 14 F2 22 01 E3
```

4. Après exécution de la séquence

a. Visualisez les contenus des emplacements mémoires

On exécute avec la commande G + l'adresse du début et l'offset de la fin du programme (l'offset de la dernière instruction) comme suit :

```
-G=073F:0000 001E
```



Remarque : LE PROGRAMME SE BLOQUE APRES L'EXECUTION EST ON PEUT PAS VISUALISER LE CONTENU DES EMLACEMENTS MEMOIRES

5. La séquence a-t-elle été exécutée correctement ? justifiez votre réponse

Non la séquence n'a pas été exécutée correctement