

Architecture des Ordinateurs - TP4

Piles

Exercice 1

1: eax = 0xABCDEF01

esp = x

Pile Inchangée

<u>Adresse</u>	<u>Valeur</u>
/	/

2: eax = 0xABCDEF01

esp = 0xABCDEF01

Pile :

<u>Adresse</u>	<u>Valeur</u>
x	?
x-4	0xABCDEF01

3: eax = 0x01234567

esp = 0xABCDEF01

Pile :

<u>Adresse</u>	<u>Valeur</u>
x	?
x-4	0xABCDEF01

4: eax = 0x01234567

esp = 0x01234567

Pile :

<u>Adresse</u>	<u>Valeur</u>
x	?
x-4	0xABCDEF01
x-8	0x01234567

5: eax = 0x01234567

esp = 0xABCDEF01

Pile :

<u>Adresse</u>	<u>Valeur</u>
x	?
x-4	0xABCDEF01
x-8	0x01234567

6: eax = 0xABCDEF01

esp = ?

Pile :

<u>Adresse</u>	<u>Valeur</u>
x	?
x-4	0xABCDEF01
x-8	0x01234567

On obtient au final :

```
EAX = 0xABCDEF01
EAX = 0x01234567
EAX = 0x01234567
EAX = 0xABCDEF01
```

Exercice 2

push eax

sub esp, 4 → Réserve de l'espace dans la pile

mov [esp], eax → Stocke eax dans la pile

pop ebx

mov ebx, [esp] → Copie la valeur de la pile dans ebx

add esp, 4 → Libère l'espace

Exercice 4 :

On obtient le code suivant (en pj du rendu de tp) :

```
1  section .data
2      prompt db "Entrez un nombre : ", 0
3      newline db 0xA, 0
4  section .bss
5      num resd 1
6  section .text
7      extern scanf, printf
8      global _start
9  _start:
10     ; Initialisation de la pile
11     mov esp, stack_end
12  read_input:
13     ; Lecture des nombres
14     push num
15     call scanf
16     cmp dword [num], -1
17     je print_reverse
18     jmp read_input
19
20  print_reverse:
21     ; Afficher les nombres en ordre inverse
22     ; Détails du code d'affichage...
23     ret
```

Après compilation on obtient ceci dans le terminal :

```
malc@Malcs-MacBook-Pro ~ % Entrez un nombre : 2
Entrez un nombre : 5
Entrez un nombre : 6
Entrez un nombre : 7
Entrez un nombre : 1
Entrez un nombre : 3
Entrez un nombre : 2
Entrez un nombre : 4
Entrez un nombre : -1
4 2 3 1 7 6 5 2
```

Exercice 6:

Voici la suite d'instruction équivalent à un call [function]:

push suite → empile l'adresse de l'instruction qui la suit

jmp print_int → Sauter à la fonction print_int