

Assembly Project

Salmi Wail Abderrahim

1_tash:

dans se tache a l'aide de l'interruption 1ch on affiche le msg oh la 1ch.... Toutes les seconde et afficher le msg programme principale dans un delai de 20 second

Intislisation de Le data et la pile et le code

```
=====
;le data
=====
data segment      Le msg sera afficher a la fin de deroutment:
    msg0 db "deroutement fait .....", 0ah, 0dh, "$"
    msg1 db 0ah, 0dh, '****programme principal en cours****$'
    msg2 db 'oh, la 1ch ..... $' :Le msg sera afficher chaque 1sec
    ;ls=1000ms la 1ch execute chack 55ms donc
    cpt dw 18;x=1000/55=18
    cm dw 20
data ends
=====
;la pile
=====
pile segment stack
    dw 128 dup(?)
    tos label word
pile ends
=====
;le code
=====
code SEGMENT
    Assume CS:code , DS:data , SS:pile
```

```
=====
;Le programme principale
=====
start:
    mov ax,data
    mov ds,ax
    mov ax,pile
    mov ss,ax
    mov sp,offset tos

    call derouter

again:mov ax,seg msg1
    mov ds,ax
    mov dx,offset msg1
    mov ah,09
    int 21h
    mov cx,0ffffh
    mov cm,20
    boucle_externe: mov si,0fffh
    boucle_interne: dec si
    jnz boucle_interne
    test cm,0ffffh,on test si cm=0 ou nn
    jz again;si zf=0 on affish le msg ****programme principal en cours****
    loop boucle_externe

    mov ax,4c00h
    int 21h
code ends
end start
```

Le cm est un coumpteur decriment au chae affich de msg 1 si il est 0 on affiche le msg 1

```

=====
;  la procedure derouter qui fait le deroutement de la routin
=====
    derouter PROC near
        push ds
        mov ax,seg new
        mov ds,ax
        mov dx, offset new
        mov ax, 251Ch
        int 21h
        pop ds
        mov dx,offset msg0
        mov ah,09
        int 21h
        ret
    derouter ENDP

=====
;  le nouvelle routine qui fait un appelle a la procedure
;  affiche chaque second
=====
    new:
        mov ax,seg cpt
        mov ds,ax
        dec cpt
        jnz sortie
        call affiche
        mov cpt,18
        dec cm;a chaque fois en affish un msg on dec cm par 1
    sortie:iret

=====
;  la procedure affiche pour afficher le msg oh, la lch....
=====
    affiche proc near
        mov ax,seg msg2
        mov ds,ax
        mov dx,offset msg2
        mov ah,09
        int 21h
        ret
    affiche endp

```

Pour meter le msg affiche chque 1sec ne doit
utiliser un compteur si cpt = 0 donc une sec est
passer donc on affiche et remettre le cpt a 18

Résultats de programme num 1

```
deroutement fait .....
```

```
****programme principal en cours****oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....  
ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....  
. oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la  
1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ...  
... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....  
****programme principal en cours****oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....  
ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....  
. oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la  
1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ...  
... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....  
****programme principal en cours****oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....  
ch ..... oh, la 1ch ..... oh, la 1ch .....
```

Dans le bonus on ajoute un compteur cmpt dans la 1ch

Dans Le data:

```
msg01 db 0ah,0dh,"recupiration fait .....","$"  
;5min=300000ms la 1ch execute chack 55ms donc  
cmpt dw 15dfh ;x=300000/55=5455  
cpt dw 3  
cm dw 20  
ancienCS dw  
ancienIP dw
```

Dans Le new routin:

```
dec cmpt  
jz near ptr derout2;si cmpt=0 (donc il past 5min) on jmp a derout2  
;=====   
; la procedure derouter qui deroute la 1ch a sont ancien routin  
;=====   
derout2 PROC near  
    push ds  
    mov ds,ancienCS  
    mov dx,ancienIP  
    mov ax, 251Ch  
    int 21h  
    pop ds  
    mov dx,offset msg01  
    mov ah,09  
    int 21h  
    jmp near ptr thend  
derout2 ENDP
```

Dans Le code principale:

```
    ; recupiration de ancian ip et cs  
;-----  
    mov ah,35h  
    mov al,1ch  
    int 21h  
    mov ancienCS,es  
    mov ancienIP,bx
```

Puis:

```
thend:mov ax,4c00h  
       int 21h  
code ends  
end start
```

```
... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....
****programme principal en cours****oh, la 1ch ..... oh, la 1ch ..... oh, la 1
ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....
. oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la
1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ...
... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....
****programme principal en cours****oh, la 1ch ..... oh, la 1ch ..... oh, la 1
ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....
. oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la
1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ...
... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....
****programme principal en cours****oh, la 1ch ..... oh, la 1ch ..... oh, la 1
ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....
. oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch ..... oh, la 1ch .....
recupiration fait .....

Program successfully executed !
Press any key to continue.
```

Résultats de la bonus part1

4_tash:

pour realiser se programme j'ai utiliser 5 compteur cpt0 qui met a 0 chaque 5 second pour afficher un msg chaque 5seconde et cpt1,cpt2,cpt3,cpt4 pour connaitre quelle msg on doit afficher

Intislisation de Le data et la pile et le code

```
=====
;le data
=====
data segment
    msg0 db "deroutement fait .....$"
    msg1 db 0ah,0dh,0ah,0dh,"tache1 en cours d'execution$"
    msg2 db 0ah,0dh,"tache2 en cours d'execution$"
    msg3 db 0ah,0dh,"tache3 en cours d'execution$"
    msg4 db 0ah,0dh,"tache4 en cours d'execution$"

    ;5s=5000ms la 1ch execute chack 55ms donc
    cpt0 dw 91d ;x=5000/55=90.9
    cpt1 dw 1
    cpt2 dw 2
    cpt3 dw 3
    cpt4 dw 4
data ends
=====
;la pile
=====
pile segment stack
    dw 128 dup(?)
    tos label word
pile ends
=====
;le code
=====
code SEGMENT
    Assume CS:code , DS:data , SS:pile
```

```
=====
;Le programme principale
=====
start:
    mov ax,data
    mov ds,ax
    mov ax,pile
    mov ss,ax
    mov sp,offset tos

    call derouter
encore: jmp encore;boucle infini

    mov ax,4c00h
    int 21h
code ends
end start
```

```

;=====
;  la procedure derouter qui fait le deroutement de la routin lch
;=====
derouter PROC near
    push ds
    mov ax,seg new
    mov ds,ax
    mov dx, offset new
    mov ax, 251Ch
    int 21h
    pop ds
    mov dx,offset msg0
    mov ah,09
    int 21h
    ret
derouter ENDP

;=====
;  le nouvelle routine qui fait un appelle a la procedure
;  affiche1 et affiche2 et affiche3 et affiche4 respictivement
;  chaque 5 second
;=====
new:
    mov ax,seg cpt0
    mov ds,ax
    dec cpt0
    jnz sortie
    mov cpt0,91d
    dec cpt1
    jnz here2
    call affiche1
    mov cpt1,4
here2:dec cpt2
    jnz here3
    call affiche2
    mov cpt2,4
here3:dec cpt3
    jnz here4
    call affiche3
    mov cpt3,4
here4:dec cpt4
    jnz sortie
    call affiche4
    mov cpt4,4

sortie:iret

```

Pour meter le msg affiche chque 5sec ne doit utiliser un compteur si cpt0 = 0 donc 5sec est passer donc on affiche et remetre le cpt a 91

Les cpt1,cpt2,cpt3,cpt4 sont des compteur se decrimente chaque 5 sec quand un de ce compteur arrives a 0 on affiche le propre msg selon l'ordre

```

;=====
;  les procedure affiche pour afficher telle tache est en cours d'execution
;=====
affiche1 proc near
    mov ax,seg msg1
    mov ds,ax
    mov dx,offset msg1
    mov ah,09
    int 21h
    ret
affiche1 endp
affiche2 proc near
    mov ax,seg msg2
    mov ds,ax
    mov dx,offset msg2
    mov ah,09
    int 21h
    ret
affiche2 endp
affiche3 proc near
    mov ax,seg msg3
    mov ds,ax
    mov dx,offset msg3
    mov ah,09
    int 21h
    ret
affiche3 endp
affiche4 proc near
    mov ax,seg msg4
    mov ds,ax
    mov dx,offset msg4
    mov ah,09
    int 21h
    ret
affiche4 endp

```


Résultats de programme num 2

```
deroutement fait .....  
  
tache1 en cours d'execution  
tache2 en cours d'execution  
tache3 en cours d'execution  
tache4 en cours d'execution  
  
tache1 en cours d'execution  
tache2 en cours d'execution  
tache3 en cours d'execution  
tache4 en cours d'execution  
  
tache1 en cours d'execution_
```

Dans le bonus on ajoute un compteur cmpt dans la 1ch

Dans Le data:

```
msg01 db 0ah,0dh,"recupiration fait .....", "$"  
;5min=300000ms la 1ch execute chack 55ms donc  
cpt dw 15dfh ;x=300000/55=5455  
;5s=5000ms la 1ch execute chack 55ms donc  
cpt0 dw 91d ;x=5000/55=90.9  
cpt1 dw 1  
cpt2 dw 2  
cpt3 dw 3  
cpt4 dw 4  
ancienCS dw  
ancienIP dw
```

Dans Le new routin:

```
dec cmpt  
jz near ptr derout2;si cmpt=0 (donc il past 5min) on jmp a derout2  
;=====  
; la procedure derouter qui deroute la 1ch a sont ancien routin  
;=====  
derout2 PROC near  
    push ds  
    mov ds,ancienCS  
    mov dx,ancienIP  
    mov ax, 251Ch  
    int 21h  
    pop ds  
    mov dx,offset msg01  
    mov ah,09  
    int 21h  
    jmp near ptr thend  
derout2 ENDP
```

Dans Le code prncipale:

```
; recupiration de ancian ip et cs  
;-----  
mov ah,35h  
mov al,1ch  
int 21h  
mov ancienCS,es  
mov ancienIP,bx
```

Puis:

```
thend:mov ax,4c00h  
       int 21h  
code ends  
end start
```

```
tache3 en cours d'execution  
tache4 en cours d'execution
```

```
tache1 en cours d'execution  
tache2 en cours d'execution  
tache3 en cours d'execution  
tache4 en cours d'execution
```

```
tache1 en cours d'execution  
tache2 en cours d'execution  
tache3 en cours d'execution  
tache4 en cours d'execution
```

```
tache1 en cours d'execution  
tache2 en cours d'execution  
tache3 en cours d'execution  
tache4 en cours d'execution
```

```
tache1 en cours d'execution  
recupiration fait .....
```

```
Program successfully executed !  
Press any key to continue.
```

Résultats de la bonus part1

FIN DU PROJET TP