TP2 – Gestion et traitement des interruptions : Correction

## Objectifs du TP :

* Comprendre les mécanismes d’interruption
* Déclenchement d’une Interruption logicielle IT DOS : codage en assembleur

Exercice 1 : Interruptions de votre machine

Etant en quantité limitée, les IRQ ont longtemps été la cause de conflits matériels, car une IRQ ne peut être partagée que sous des conditions très strictes. Afin de vous faire une idée de l'utilisation des interruptions vous irez sur votre machine dans les **Propriétés systèmes,** puis en ouvrant le **Gestionnaire de périphériques**. Recherchez par exemple votre clavier dans la liste, puis choisissez **Propriétés** et regardez l'onglet **Ressources**.

Exercice 2 : Table des vecteurs d’interruptions de votre machine

On veut visualiser le contenu de la table des vecteurs d’IT.  
Prendre l’exemple de l’interruption 21h interruption du DOS.  
Visualisez la mémoire à l’adresse 0000:0084, la valeur du vecteur d’IT.   
(Exemple sur mon PC je vois : 00 02 00 F4 ce qui correspond à l’adresse F400:0200)

Allez à cette adresse (en mettant le registre CS à la valeur F400 et IP à la valeur 0200) et vérifiez bien que le code de la procédure de l’interruption logicielle 21h. Le seul moyen de vérifier est de constater qu’elle se termine bien par un IRET.

Exercice 3 : Déclenchement d’une interruption logicielle

Rappel **:**

Lorsqu’une IT est déclenchée le micro-processeur :

* empile les indicateurs du registre d’état
* désactive les IT matérielles
* empile CS
* empile IP

Il exécute ensuite la routine d’interruption. Cette routine se termine par un IRET qui a pour effet de :

• dépiler IP

• dépiler CS

• dépiler les indicateurs

Pour déclencher une interruption en assembleur il suffit d’invoquer le numéro de l’interruption par l’instruction INT.

Affichage de l’heure courante du PC :

Vous allez invoquer dans une boucle, l’interruption qui donne l’heure courante de votre PC et vous afficherez ce résultat avec le format suivant :

*L heure actuelle est : 12h01m36s11*

*L heure actuelle est : 12h01m36s43*

*L heure actuelle est : 12h01m37s15*

**Interruption** de l’horloge est INT 21h

**Entrée** : AH=2Ch

**Sortie** : CH = hour CL = minute DH = second DL = 1/100 seconds  
Solution :

Ecrivez le code assembleur correspondant.   
Vous constaterez que les valeurs affichées ne sont pas qu format HH :MM :SS :CC.   
Vous écrirez donc une procédure qui permet la conversion des valeurs retournées par l’interruption en des valeurs affichables :  
Quelle est la durée moyenne d’exécution d’un passage dans la boucle ?

; Correction Exo1\_IntHorloge.asm

STACK SEGMENT

DW 256 DUP (0)

base:

STACK ENDS

DATA SEGMENT

msg DB 'L heure actuelle est : $'

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STACK

;Converti le premier parametre (valeur entre 0 et 99) en 2 caracteres ASCII

;IN : parametre 1 : la valeur a convertir en texte

; : parametre 2 : vide, il sert juste pour le retour

; OUT : parametre 1 : la valeur ASCII de l'unite

; : parametre 2 : la valeur ASCII de la dizaine

**convertionValeurASCII PROC NEAR**

PUSH BP

PUSH AX

PUSH BX

MOV BP, SP

MOV AX, [BP+10]

MOV BL, 10

DIV BL

; AH Reste, AL Quotient

MOV BL, AH

MOV BH, 0

ADD BX, 48

MOV [BP+10], BX

MOV AH, 0

ADD AX, 48

MOV [BP+8], AX

POP BX

POP AX

POP BP

RET

**convertionValeurASCII ENDP**

**debut:**

MOV AX, Data

MOV DS, AX

MOV AX, Stack

MOV SS, AX

MOV SP, base

MOV SI, 10

; Affichage

MOV DX, OFFSET msg

MOV AH, 9

INT 21h

boucle:

MOV DH, 0

MOV DL, 22

MOV BH, 0

MOV AH, 2

INT 10h

; Appel de l'interruption

MOV AH, 2Ch

INT 21h

; CH : Heure, CL Minutes, DH Secondes, DL centiemes de s.

MOV BX, DX ; sauvegarde de DX qui contient les secondes et les 100eme

; gestion des heures

MOV DH,0

MOV DL, CH

PUSH DX

PUSH DX ; sert juste a avoir un 2eme param de retour

CALL convertionValeurASCII

POP DX

MOV AH, 02

INT 21h

POP DX

MOV AH, 02

INT 21h

MOV DL, 'h'

MOV AH, 02

INT 21h

; gestion des minutes

MOV DH,0

MOV DL, CL

PUSH DX

PUSH DX ; sert juste a avoir un 2eme param de retour

CALL convertionValeurASCII

POP DX

MOV AH, 02

INT 21h

POP DX

MOV AH, 02

INT 21h

MOV DL, 'm'

MOV AH, 02

INT 21h

; gestion des secondes

MOV DH,0

MOV DL, BH

PUSH DX

PUSH DX ; sert juste a avoir un 2eme param de retour

CALL convertionValeurASCII

POP DX

MOV AH, 02

INT 21h

POP DX

MOV AH, 02

INT 21h

MOV DL, 's'

MOV AH, 02

INT 21h

; gestion des 100eme de secondes

MOV DH,0

MOV DL, BL

PUSH DX

PUSH DX ; sert juste a avoir un 2eme param de retour

CALL convertionValeurASCII

POP DX

MOV AH, 02

INT 21h

POP DX

MOV AH, 02

INT 21h

;gestion de la boucle

MOV DL, 10

MOV AH, 02

INT 21h

MOV DL, 13

INT 21h

DEC SI

CMP SI,0

JNE boucle

fin:

MOV AH, 4CH

INT 21H

**CODE ENDS**

**END debut**

Exercice 3 : création d’une routine d’ Interruption logicielle

Vous allez invoquer votre propre routine d’interruption en utilisant un numéro d’interruption laissé libre pour l’utilisateur.   
Vous choisirez l’interruption n° 150.   
Le programme main met en place la routine d’IT dont le numéro sera 150 puis fait une boucle qui teste la valeur d’un compteur qui mémorise le nombre d’appel a l’IT avant de sortir.

Il faut rappeler les points suivants :

les fonctions du DOS qui permettent d’installer une procédure d’IT et de récupérer l’adresse d’un handler d’IT

**INT 21h**

**Fonction 25h** qui installe une procédure d’IT dans la table des vecteurs d’IT.

*entrée* :

AH <- numéro de la fonction

AL <- numéro de l’IT

DX <- offset de la procédure d’IT

DS <- segment de la procédure d’IT

*sortie :*

rien

**INT 21h**

**Fonction 35h** qui récupère l’adresse d’une procédure d’IT dans la table des vecteurs d’interruption

entrée :

AH <- numéro de la fonction

AL <- numéro de l’IT

sortie :

BX -> offset de la procédure d’IT

ES -> Segment de la procédure d’IT

Voici le pseudo-code des trois procédures à coder :

**Main**

Appel de la procédure Installation\_IT-150

Compteur <- 0

iterer

imprime M

invoque l’interruption numéro 150

teste un compteur

sort si compteur = 10

al <- compteur

affiche al

fin iterer

**Votre\_procedure\_dIT\_150**

compteur ++

imprime ‘I’

**Installation\_IT\_150**

AH<-25h

AL <- 150

DX <- offset de votre\_procédure\_dIT\_150

DS <- segment de votre\_procédure\_dIT\_150

Invoquer l’interruption 21h

Ne pas oublier qu’une procédure d’IT doit se terminer par IRET.  
Pour récupérer la valeur de l’offset d’une procédure en assembleur on utilise d’instruction OFFSET.  
Pour récupérer la valeur de segment d’une procédure en assembleur il suffit d’utiliser le nom du segment de la procédure (voir solution).

Solution :

**; Exo2\_IT\_Logicielle**

PILE SEGMENT STACK

DB 256 DUP(?)

PILE ENDS

DONNEES SEGMENT

compteur db 0

DONNEES ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DONNEES

**EXO2 PROC FAR**

PUSH DS

MOV AX,0

PUSH AX

MOV AX,DONNEES

MOV DS,AX

call INSTALLATION\_IT\_150

mov compteur,0

iterer:

mov dl, 'M'

call cout

int 150

cmp compteur, 10

je fin

mov dl, compteur

add dl, '0'

call cout

; carriage return

mov dl,0Dh

call cout

jmp iterer

fin:

RET

**EXO2 ENDP**

**VOTRE\_PROCEDURE\_DIT\_150 PROC NEAR**

Inc compteur

Mov dl,'I'

Call cout

iret

**VOTRE\_PROCEDURE\_DIT\_150 ENDP**

**INSTALLATION\_IT\_150 PROC NEAR**

Mov AH, 25h

Mov AL,150

Mov DX, OFFSET VOTRE\_PROCEDURE\_DIT\_150

; on met dans DS la valeur du segment de la procédure

MOV SI,CODE ; SI prend la valeur du segment qui contient

; la procédure

PUSH SI ; on met SI sur la pile

POP DS ; on dépile dans DS

Int 21h

ret

**INSTALLATION\_IT\_150 ENDP**

**COUT PROC near**

mov AH, 02

int 21h

ret

**COUT ENDP**

**CODE ENDS**

END EXO2