

Interrupciones en ensamblador¹

Las interrupciones son habilidades de comunicación entre hardware y software con el programa que se está desarrollando que le permiten ampliar su capacidad al manejar hardware a través del sistema operativo, o recibir datos o mensajes del sistema operativo, avisar que el programa principal ha terminado, establecer la comunicación hacia algún dispositivo de entrada o salida como la pantalla, el teclado, el mouse o una impresora.

Las interrupciones en un sistema embebido permiten además avisarle al programa principal de eventos que estén sucediendo en el exterior, como el estado de la temperatura del sistema, de la batería, incluso si una determinada terminal se encuentra en uno o en cero.

Es muy importante que un programa de cómputo posea la capacidad de interactuar con el usuario final, por lo menos a través de la pantalla y el teclado en el caso de una computadora personal. En ensamblador utilizamos el servicio de interrupciones del sistema operativo para comunicarnos con el hardware.

Es así que podemos programar rutinas que muestren en pantallas resultados o, incluso, gráficos que muestren al usuario diversas informaciones útiles; también es posible hacer subrutinas para que se introduzcan datos como cadenas de texto o valores numéricos e, incluso, comandos, a través del teclado y el ratón.

Con las interrupciones además podemos comunicarnos con otros dispositivos de hardware unidos a la computadora, como las impresoras, escáners, puertos seriales y paralelos.

Además, el hardware de la computadora posee interrupciones de software propias definidas por un número específico, por ejemplo, la interrupción **10h** maneja los servicios de video, la **16h** el teclado, la **17h** la impresora. Una de las interrupciones más utilizada es la **21h**, que es la interrupción a los servicios del sistema operativo. La interrupción **21h** es muy importante porque entre todas las funciones que posee incluye funciones de manejo del hardware, teclado, video, archivos, terminación de programa, incluso la interpretación de comandos del sistema operativo sin perjuicio de las interrupciones propias del hardware. Es decir, que hay dispositivos que podemos controlar desde su propia interrupción o a través de las del sistema operativo.

¹ Documento elaborado por el Dr. Carlos Jiménez Gallegos.

Para comunicar un programa con el exterior se hace uso de los servicios de interrupción. Hasta el momento hemos utilizado sólo la interrupción del sistema operativo, la llamada interrupción 21H.

Hemos empleado el servicio *09H* de la interrupción 21H para enviar una cadena a la pantalla y el servicio *4CH* de la misma interrupción 21H, que es la función de terminar programa y regresar el control al sistema operativo.

De toda la gama de interrupciones nos interesan, además de la 21H, la 10H que maneja el video, la 16H que maneja el teclado y podría ser muy útil usar la **33H**, que se encarga del teclado. En el apéndice C del libro de Irvine (Irvine, 2007, págs. 650-658) hay unas tablas que resumen estas 4 interrupciones mencionadas. Por el momento, sólo mencionaremos los servicios más comunes a los que accedemos para darle una interactividad y presentación aceptable a nuestros programas.

De la interrupción 10H usaremos, por ejemplo, el servicio 0H que es **establecer el modo de video**, 2H para establecer **la posición del cursor en la pantalla**, 9H con la que **escribimos un carácter ASCII en la posición actual del cursor**. En el sitio <http://www.ctyme.com/intr/int-10.htm> podrás encontrar una lista muy detallada de todas las interrupciones y servicios de la PC (Brown, 2014).

Hay servicios que realizan la misma acción. Por ejemplo, la interrupción 16H tiene el servicio 10H que es equivalente al servicio 1H de la interrupción 21H. Por lo que su uso es indistinto con una u otra interrupción.

Veamos este **primer ejemplo** que hace uso del teclado y la pantalla para mostrar el resultado de una operación: multiplicar 2 cifras. Recuerde que al introducir una cifra por teclado ésta entra a la memoria como su valor ASCII por lo que será necesario ajustar restándole 30H (39H es el '9' en ASCII y 30H es el '0' ASCII).

```

.MODEL SMALL           ;Modelo pequeño

.STACK 100h           ;Segmento de pila 256 posiciones

CR EQU 13             ;Retorno de carro
LF EQU 10             ;nueva línea

.DATA                 ;Segmento de datos

TEXT01 DB 'Operaciones de dos cifras ',LF,"",CR,$
    
```

TEXT02 DB CR, LF, 'Proporcione el primer n',163,'mero (dos cifras) x= \$'

TEXT03 DB CR, LF, LF, 'Proporcione el segundo (dos cifras) y= \$'

TEXT04 DB CR, LF, LF, 'Los resultados x/y x+y y xy son: \$'

.CODE ;Código del programa

; ***** Texto de presentación ****

MOV AX,@DATA ; Cargar datos

MOV DS,AX ; Cargar la dirección en el segmento de datos

LEA DX,TEXT01 ; Carga el texto

MOV AH,9h ; imprimir en pantalla

INT 21h

; Llama al sistema operativo

; ***** AQUI SE INICIA EL PROGRAMA *****

; ** 1) Pedir primer valor

LEA DX,TEXT02 ; Carga el texto

MOV AH,9h ; imprimir en pantalla

INT 21h ; Llama al sistema operativo

; ** 1.a X primer caracter (decenas) y multiplica por 10

MOV AH,01 ; Leer del teclado 1 caracter

INT 21H

SUB AL,30H ; obtiene valor en decimal

MOV CL,0AH

```
MUL CL                ; multiplica por 10
MOV BH,AL             ; se guarda en BH

; ** 1.b X segundo caracter (unidades) y lo suma al anterior

MOV AH,01             ;Leer del teclado 1 caracter
INT 21H
SUB AL,30H            ;obtiene valor en decimal
ADD BH,AL             ;se suma a las decenas

; ** 2) Pedir segundo valor

LEA DX,TEXT03         ;Carga el texto
MOV AH,9h             ;imprimir en pantalla
INT 21h               ;Llama al sistema operativo

; ** 2.a Y primer caracter (decenas) y multiplica por 10

MOV AH,01             ; Leer del teclado 1 caracter
INT 21H
SUB AL,30H            ; obtiene valor en decimal
MOV CL,0AH
MUL CL                ; multiplica por 10
MOV BL,AL             ; se guarda en BL

; ** 2.b Y segundo caracter (unidades) y lo suma al anterior

MOV AH,01             ; Leer del teclado 1 caracter
INT 21H
SUB AL,30H            ; obtiene valor en decimal
ADD BL,AL             ; se suma a las decenas
```

; *** en este punto BH tiene un operando y BL el otro

; ** 3) Operaciones

; ** 3.1) multiplicación

MOV AL,BH ; se carga el primer operando
MOV CL,BL ; segundo operando
MUL CL ; multiplicación AX=AL*CL
PUSH AX ; el resultado se guarda en la pila

; ** 3.2) suma

MOV AX,0000H ; se borra AX
MOV AL,BH ; se carga primer operando
ADD AL,BL ; suma AL=AL+BL
PUSH AX ; se almacena el resultado en la pila

; ** 3.3) división (entera)

MOV AX,0000H ; se borra AX
MOV AL,BH ; primer operando
MOV CL,BL ; segundo operando
DIV CL ; división AX/CL (AL) tiene cociente
MOV AH, 00H ; borramos residuo (AH)
MOV BX,AX ; Guardar resultado de la división
; se guarda en BX pues AX se usa para
; mostrar el texto 4

; ** 4) proceso de ajuste para pantalla

; ** 4.1) letrero

LEA DX,TEXTO4 ;Carga el texto

```
MOV AH,9h      ;imprimir en pantalla
INT 21h        ;Llama al sistema operativo

MOV AX, BX      ;volver a colocar el resultado en AX

; **4.2) resultado entero de la división
MOV DX,0000H
MOV CX,100
DIV CX          ; AX/CX= AX/100 (obtenemos millares y centenas)
AAM             ; ajuste ASCII de las centenas
                ; AH millares AL centenas
ADD AX,3030H    ; ajustes ASCII centenas
MOV BX,AX       ; BH millares BL centenas
MOV CX,DX       ; guardar en CX el residuo

MOV DL,BH       ; millares
MOV AH, 06H     ;servicio 06H imprimir ASCII que está en DL
INT 21H
MOV DL,BL       ; centenas
MOV AH, 06H     ;06H imprimir ASCII que está en DL
INT 21H

XCHG AX,CX      ;intercambiar registros AX con CX
AAM             ; Ajuste decenas y unidades
ADD AX, 3030H   ; Ajuste ASCII DECENAS Y CENTENAS
                ; AH decenas AL unidades

MOV BX,AX       ; guardar en BX

MOV DL,BH       ; Decenas
MOV AH, 06H     ;06H imprimir ASCII que está en DL
```

```
INT 21H
MOV DL,BL ;unidades
MOV AH, 06H ;06H imprimir ASCII que está en DL
INT 21H

MOV DL,20H
MOV AH, 06H ;06H imprimir un espacio
INT 21H

; **4.3) resultado de la suma
; ** se repite el proceso con el resultado de la suma

POP AX ; se recupera el resultado de la suma

MOV DX,0000H
MOV CX,100
DIV CX
AAM
ADD AX,3030H
MOV BX,AX
MOV CX,DX

MOV DL,BH
MOV AH, 06H
INT 21H
MOV DL,BL
MOV AH, 06H
INT 21H

XCHG AX,CX
AAM
ADD AX, 3030H
```

MOV BX,AX

MOV DL,BH

MOV AH, 06H

INT 21H

MOV DL,BL

MOV AH, 06H

INT 21H

MOV DL,20H

MOV AH, 06H

INT 21H

; **4.4) resultado de la multiplicación

; ** se muestra el último resultado

POP AX ; se recupera de la pila

MOV DX,0000H

MOV CX,100

DIV CX

AAM

ADD AX,3030H

MOV BX,AX

MOV CX,DX

MOV DL,BH

MOV AH, 06H

INT 21H

MOV DL,BL

MOV AH, 06H

```
INT 21H

XCHG AX,CX
AAM
ADD AX, 3030H

MOV BX,AX

MOV DL,BH
MOV AH, 06H
INT 21H
MOV DL,BL
MOV AH, 06H
INT 21H

MOV DL,20H
MOV AH, 06H
INT 21H

; ***** AQUI SE FINALIZA EL PROGRAMA *****

MOV Ax,4c00h           ;Función 4ch
INT 21h

END                   ;Fin del programa
```

Programa 1

Note como la programación ha ganado presentación, aunque es aún rígida, por ejemplo, sólo acepta dos cifras y la división es entera. Una instrucción nueva que aparece es *AAM* que ajusta los valores del registro *AX* a valores ASCII. Como última observación, podrá verificar que sólo se utilizaron servicios de la interrupción 21H, es decir, que todas las peticiones del teclado se llevaron a cabo a través del sistema operativo (MS DOS emulado si está usando **Emu 8086**).

Pasemos a analizar un **segundo programa** que haga uso de las opciones de video, para agregarle colorido a la presentación. Note como el tamaño del código ha crecido y claramente resaltan zonas de código repetitivo. Más adelante mejoraremos la implementación al crear librerías.

```

.MODEL SMALL                ;Modelo pequeño

.STACK 100h                 ;Segmento de pila 256 posiciones

        CR EQU 13           ;Retorno de carro
        LF EQU 10           ;nueva línea

.DATA                        ;Segmento de datos

TEXTO1 DB 'Interrupci',162,'n de video$'
TEXTO2 DB 'Men',163,': $'

.CODE                        ; Código del programa

; ***** Texto de presentación *****
        MOV AX,@DATA        ; Cargar datos

        MOV DS,AX           ; Cargar la dirección en el segmento de datos
        LEA DX,TEXTO1       ; Carga el texto
        MOV AH,9h           ; imprimir en pantalla
        INT 21h

        MOV AH,08h          ; servicio 08H: esperar una tecla
        INT 21h

; ***** AQUI SE INICIA EL PROGRAMA *****

```

; 1) *** BORRAR LA PANTALLA (poniendo otro fondo)

```
MOV AL, 00H    ; Número de líneas a borrar
                ; 00H es borrar todo
MOV BH, 010H   ; atributo caracter
MOV CX, 0000H  ; CH renglón, CL columna inicial
                ; comenzar en (0,0)
MOV DH, 25     ; terminar en la fila
MOV DL, 80     ; terminar en la columna
MOV AH, 06H    ; desplazar
INT 10H        ; interrupción de video
```

;2) *** Dibujar un Marco en pantalla ***

; el caracter CDh es una doble línea

; colocar el cursor en el inicio

```
MOV BH, 0H
MOV DH, 0      ; renglón
MOV DL, 0      ; columna
MOV AH, 02H    ; colocar cursor
INT 10H
```

;imprimir línea superior

```
MOV AL, 0CDH   ; caracter
MOV BH, 00H    ; página de video
MOV BL, 07H    ; atributo del caracter
MOV CX, 80     ; repeticiones
MOV AH, 09H    ; escribir con atributo
INT 10H
```

; imprimir línea inferior

```
MOV BH,0H
MOV DH,24    ; renglón
MOV DL,0     ; columna
MOV AH,02H   ; colocar cursor
INT 10H
MOV AL,0CDH  ; caracter
MOV BH,00H   ; página de video
MOV BL,01H   ; atributo del caracter
MOV CX, 80   ; repeticiones
MOV AH,09H   ; escribir con atributo
INT 10H
```

; imprimir línea izquierda

```
MOV CX,23
```

lazo01:

```
MOV BH,0H
MOV DL,0     ; columna
MOV DH,CL    ; renglón
MOV AH,02H   ; colocar cursor
INT 10H

MOV AL,186   ; caracter
MOV BH,00H   ; página de video
MOV BL,01H   ; atributo del caracter
PUSH CX      ; se guarda la cuenta del ciclo
MOV CX, 1    ; repeticiones del caracter
MOV AH,09H   ; escribir con atributo
INT 10H
POP CX
LOOP lazo01
```

; imprimir línea derecha

MOV CX,23

lazo02:

MOV BH,0H

MOV DL,79 ; columna

MOV DH,CL ; renglón

MOV AH,02H ; colocar cursor

INT 10H

MOV AL,186 ; caracter

MOV BH,00H ; página de video

MOV BL,01H ; atributo del caracter

PUSH CX ; se guarda la cuenta del ciclo

MOV CX, 1 ; repeticiones del caracter

MOV AH,09H ; escribir con atributo

INT 10H

POP CX

LOOP lazo02

; imprimir esquinas

MOV BH,0H

MOV DL,0 ; columna

MOV DH,0 ; renglón

MOV AH,02H ; colocar cursor

INT 10H

MOV AH,09H

MOV AL,201 ; esquina superior izquierda

MOV BX,0001H

```
MOV CX,1
```

```
INT 10H
```

```
MOV BH,0H
```

```
MOV DL,79      ; columna
```

```
MOV DH,0       ; renglón
```

```
MOV AH,02H     ; colocar cursor
```

```
INT 10H
```

```
MOV AH,09H
```

```
MOV AL,187     ; esquina superior derecha
```

```
MOV BX,0001H
```

```
MOV CX,1
```

```
INT 10H
```

```
MOV BH,0H
```

```
MOV DL,0       ; columna
```

```
MOV DH,24      ; renglón
```

```
MOV AH,02H     ; colocar cursor
```

```
INT 10H
```

```
MOV AH,09H
```

```
MOV AL,200     ; esquina inferior izquierda
```

```
MOV BX,0001H
```

```
MOV CX,1
```

```
INT 10H
```

```
MOV BH,0H
```

```
MOV DL,79      ; columna
```

```
MOV DH,24      ; renglón
```

```
MOV AH,02H     ; colocar cursor
```

```
INT 10H
```

```
MOV AH,09H
MOV AL,188    ; esquina inferior derecha
MOV BX,0001H
MOV CX,1
INT 10H
```

```
; *** colocar letrero al centro ***
```

```
; colocando cursor
```

```
MOV BH,0H
MOV DL,38    ; columna
MOV DH,10    ; renglón
MOV AH,02H   ; colocar cursor
INT 10H
```

```
; letrero
```

```
LEA DX,TEXT02    ; Carga el texto
MOV AH,9h        ; imprimir en pantalla
INT 21h
```

```
; espera una tecla antes de terminar
```

```
MOV AH,08h      ; servicio 08H: esperar una tecla
INT 21h
```

```
; ***** AQUI SE FINALIZA EL PROGRAMA *****
```

```
MOV Ax,4c00h    ;Función 4ch
INT 21h
```

END ;Fin del programa

Programa 2