

Seminario de Solución de problemas de Traductores de Lenguajes I

Centro Universitario de Ciencias Exactas en
ingenierías

Universidad de Guadalajara



Maestro: Tonatiuh Hernandez Casas

Juan Antonio Pérez Juárez
Código: 215660996
Carrera: INCO

Actividad 5: Parte I

Actividad 5: Parte II

Escriba un programa en ensamblador que haga lo siguiente:

- Borrar pantalla
- Configurar el curso en una posición específica
- Muestre un mensaje en pantalla

```
Unset
.MODEL SMALL
.STACK 100H
.DATA
    ;Mensaje
    MSG DB "Este programa fue Muy dificil de realizar, profe... Plis,
piedad"

.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    ;CLS
    MOV AH, 06H
    MOV AL, 0
    MOV BH, 02H
    MOV CH, 0
    MOV CL, 0
    MOV DH, 24
    MOV DL, 79
    INT 10H

    MOV AH, 02H    ;Posicion del Cursos
    MOV BH, 0
    MOV DH, 12
    MOV DL, 15
    INT 10H

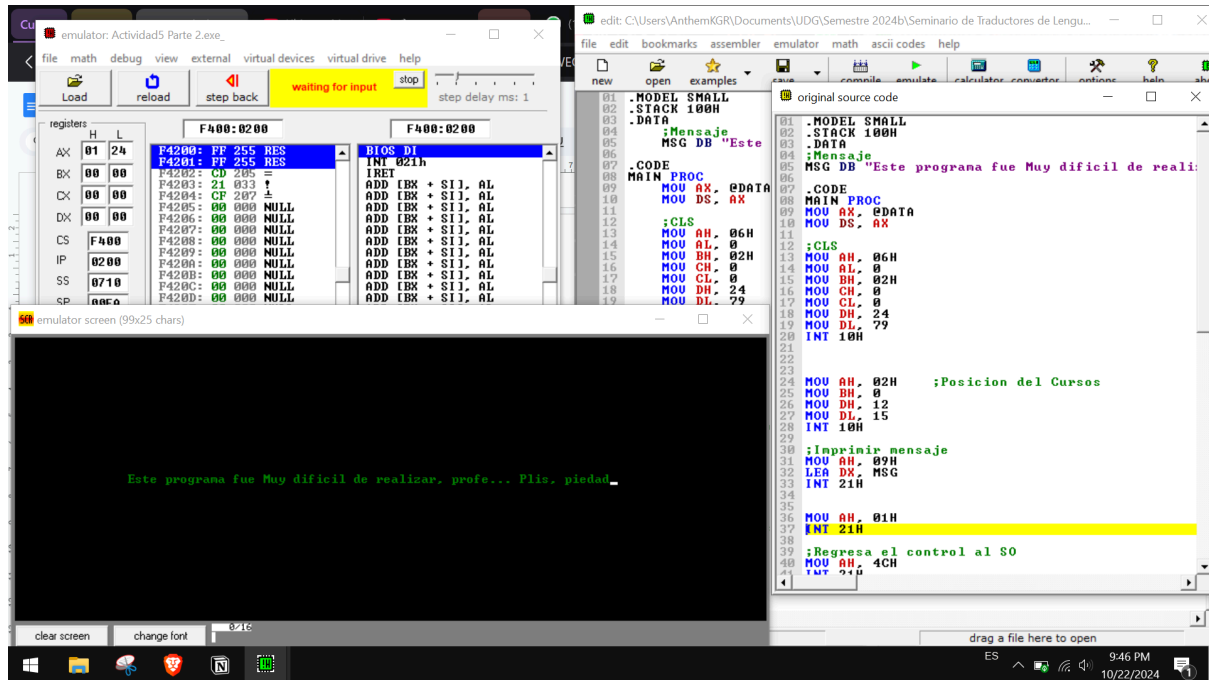
    ;Imprimir mensaje
    MOV AH, 09H
    LEA DX, MSG
    INT 21H
```

```

MOV AH, 01H
INT 21H

;Regresa el control al SO
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN

```



Actividad 5: Parte III

Explique que hacen las siguientes funciones o servicios de las interrupciones 21H y 10H

- 4C00H de la 21H

Es la función para terminar el programa y volver al DOS

El "00H" es el código de retorno (exit code) que indica que el programa terminó correctamente

Es como decirle al sistema "Cierra todo, ya se terminó el programa"

- 09H de la 21H

Sirve para imprimir una cadena de caracteres (string) en pantalla

La cadena debe terminar con el carácter '\$'

El registro DX debe contener la dirección de la cadena a imprimir

- 0AH de la 21H

Lee una cadena de caracteres desde el teclado.

El servicio 0Ch de la INT 21h borra el buffer del teclado e invoca un servicio de entrada de carácter de los explicados más arriba (01h, 06h, 07h, 08h) o de entrada de cadenas que se verá seguidamente (0Ah). El registro AL se utiliza para seleccionar el servicio a invocar y devuelve el valor propio de dicho servicio.

- 02H de la 10H

Posiciona el cursor en la pantalla

DH = número de fila (0-24)

DL = número de columna (0-79)

BH = número de página de video (normalmente 0)

Realiza operaciones de posicionamiento de cursos en pantalla.

- 06H de la 10H

Desplaza la ventana hacia arriba (scroll up) o limpia la pantalla

AL = número de líneas a desplazar (0 = limpiar toda la ventana)

BH = atributo a usar en las líneas nuevas (color)

CH, CL = fila, columna de la esquina superior izquierda

DH, DL = fila, columna de la esquina inferior derecha

Se usa comúnmente para limpiar la pantalla poniendo AL = 0

Analiza el siguiente código:

```
Unset
PAGE 60,132
TITLE PROG8.EXE
.MODEL SMALL
.STACK 64
;-----
.DATA
ENTRA LABEL BYTE
LONMAX DB 20
LONREAL DB ?
INTROD DB 21 DUP ( ' ')
MEN DB 'INTRODUCIR NOMBRE: ' , '$'
;-----
.CODE
BEGIN PROC FAR
MOV AX, @DATA
MOV DS, AX
OTRO: CALL PANT0
MOV DX, 0502H
CALL CURS0
CALL DESPL0
CALL TECLA0
```

```
CMP LONREAL,00
JE SALIR
CALL CAMPA
CALL CENTRAR
JMP OTRO
SALIR: MOV AX,4C00H
INT 21H
BEGIN ENDP
```

```
;-----
DESPL0 PROC NEAR
MOV AH,09H
LEA DX, MEN
INT 21H
RET
DESPL0 ENDP
```

```
;-----
TECLA0 PROC NEAR
MOV AH,0AH
LEA DX, ENTRA
INT 21H
RET
TECLA0 ENDP
```

```
;-----
CAMPA PROC NEAR
MOV BH,00
MOV BL, LONREAL
MOV INTROD[BX],07H
MOV INTROD[BX+1],'$'
RET
CAMPA ENDP
```

```
;-----
CENTRAR PROC NEAR
MOV DL, LONREAL
SHR DL,1
NEG DL
ADD DL,40
MOV DH,12
CALL CURS0
MOV AH,09H
LEA DX, INTROD
INT 21H
RET
CENTRAR ENDP
```

```
;-----
PANT0 PROC NEAR
MOV AX,0600H
```

```

MOV BH,30
MOV CX,0000
MOV DX,184FH
INT 10H
RET
PANT0 ENDP
;-----
CURS0 PROC NEAR
MOV AH,02H
MOV BH,00
INT 10H
RET
CURS0 ENDP
;-----
END BEGIN

```

Así que vamos parte por parte para tratar de desmenuzar.

new	open	examples	save	compile	emulate
01	PAGE 60,132				
02	TITLE PROG8.EXE				
03	.MODEL SMALL				
04	.STACK 64				
05	;				
06	DATA				

Define el tamaño del programa y asigna el tamaño de la pila en 64.

06	.DATA				
07	ENTRA LABEL BYTE				
08	LONMAX DB 20				
09	LONREAL DB ?				
10	INTROD DB 21 DUP <' '>				
11	MEN DB 'INTRODUCIR NOMBRE: ' , '\$'				
12	;				

Sección que podía ser considerada de datos y se declaran las variables.

```

;-----
.CODE
BEGIN PROC FAR
MOV AX, @DATA
MOV DS, AX
OTRO: CALL PANT0
MOV DX, 0502H
CALL CURS0
CALL DESPL0
CALL TECLA0
CMP LONREAL, 00
JE SALIR
CALL CAMPA
CALL CENTRAR
JMP OTRO
SALIR: MOV AX, 4C00H
INT 21H
BEGIN ENDP

;-----
DESPL0 PROC NEAR
MOV AH, 09H
LEA DX, MEN
INT 21H
RET
DESPL0 ENDP

;-----
TECLA0 PROC NEAR
MOV AH, 0AH
LEA DX, ENTR0
INT 21H
RET
TECLA0 ENDP

```

A partir de aquí comienza con la sección del código.

Limpia la pantalla.

Posiciona el cursor e imprime el mensaje de “Introducir Nombre:”

```

;-----
CENTRAR PROC NEAR
MOV DL, LONREAL
SHR DL, 1
NEG DL
ADD DL, 40
MOV DH, 12
CALL CURS0
MOV AH, 09H
LEA DX, INTROD
INT 21H
RET
CENTRAR ENDP

```

Aquí hay un pedazo de código que me gustó, que lo que hace es emitir un sonido, no sabía que se podía hacer eso, pero es muy lógico que se pueda.

```

;-----
CURS0 PROC NEAR
MOV AH, 02H
MOV BH, 00
INT 10H
RET
CURS0 ENDP

```

Limpia la pantalla con un color específico

```
;-----  
TECLA0 PROC NEAR  
MOV AH,0AH  
LEA DX,ENTRA  
INT 21H  
RET  
TECLA0 ENDP  
;-----
```

Lee el texto desde el teclado y carga la dirección en la memoria.

Reflexión

Woah, esta actividad, de no ser por que hay miles de repositorios y forums como reddit para poder extraer el código de otras personas, no la habría terminado a tiempo y he de reconocer que en la parte I no fuí a clases, en general no he podido asistir por unos problemas personales y algunos encargos adicionales en mi trabajo, así que la parte I no la hice y espero que en las siguientes actividades y en las siguientes partes poder hacerlas como corresponde así como dedicarle el tiempo necesario para poder aprender.

Porqué me gustaría poder entender el funcionamiento de ensamblador para poder hacer algunos proyectos personales que tengo en mente.

Y el programa de la parte III se me hizo realmente genial, genuinamente lo analicé y traté de entender todo lo que pude, y es bastante diferente de cómo funcionan los lenguajes de programación de alto nivel, hice lo mismo en python y es hasta gracioso que lo mismo se pueda hacer en 20 líneas de código (Que se note que no se optimizar código). Pero realmente me gustó esta actividad y le extiendo una sincera disculpa por no dedicar el tiempo necesario a su clase profesor.

Bibliografía:

<https://www.cartagena99.com/recursos/alumnos/ejercicios/210428171148-Laboratorio%20Practica3.pdf>

Funciones del DOS. (n.d.).

http://arantxa.ii.uam.es/~gdrivera/labetcii/int_dos.htm

https://gc.scalahed.com/recursos/files/r161r/w25840w/S5_InterrupcionesEnEnsamblador_Semana_5REV.pdf

Facultad de Ingeniería UNAM. (2021). Apuntes de Lenguaje Ensamblador.

<http://dctrl.fi-b.unam.mx/>

Universidad Politécnica de Madrid. (2020). Manual de Programación en Ensamblador. <http://www.datsi.fi.upm.es/docencia/>

Intel Corporation. (2021). Intel® 64 and IA-32 Architectures Software Developer's Manual. <https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html>

Tutorials Point. (2021). Assembly Programming Tutorial. https://www.tutorialspoint.com/assembly_programming/