



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO SUPERIOR PROGRESO

**PROGRAMA ACADÉMICO DE INGENIERÍA EN
SISTEMAS COMPUTACIONALES**

ASIGNATURA

Lenguaje de interfaz.

DOCENTE

Edgar Alejandro Sagundo Duarte.

TRABAJO

Macros y procedimientos.

PRESENTA

Manuel Ricardo Uc Nicoli 04190032

Progreso Yucatán, 28 de abril de 2022.

Actividad 2 – Macros y procedimientos.

Desarrollar un programa en ensamblador que permita limpiar la pantalla y mostrar una bandera de algún país, indicando abajo el nombre de la bandera.

Objetivo: Resolver problemas reales empleando macros y procedimientos en ensamblador

Rubrica.

Rúbrica	Excelente	Muy bien	Regular	Mal
Cantidad y funcionalidad	Todos los ejercicios estuvieron correctos sin errores	Del 90 al 50% de los ejercicios estuvieron correctos con advertencias	Menos del 50% estuvieron correctos con 1 error	No realizó los ejercicios o el 100% esta incorrecto
Entrega	Entrego en tiempo y forma	Entrego en tiempo pero no en forma		No entrego a tiempo y sin forma

Segmentación de código:

BANDERA.ASM

;@Manuel Uc Nicoli

;Desarrollar un programa en ensamblador que permita limpiar la pantalla

;y mostrar una bandera de algún país, indicando abajo el nombre de la bandera.

STACK SEGMENT STACK

DW 64 DUP(?)

STACK ENDS

include "Macros-procedimientos.asm" ;incluimos nuestras macros de la misma carpeta en otro caso especificar ruta - C:\ruta\ruta.extension

;--- PILA ----

DATA SEGMENT

txt1 db 10,13,'Programa finalizado',10,13,'\$'

txtB db 10,13,'Bandera de Ucrania',10,13,'\$'

DATA ENDS

;--- DATOS ---

CODE SEGMENT

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

INICIO: mov AX, DATA

mov DS, AX

BANDERA:

Call LimpiarP

BanderaSuperior

BanderaInferior

Puntero 9

Mensaje txtB

Call Fin

LimpiarP proc near ;limpiar pantalla.

 ;funcion para hacer scroll tambien con 7h

 mov ax,0600h

 mov bh,07h;atributos de color fondo y texto

 mov CX,0000h;fila inicial en ch, columna inicial en cl

 mov DX,0000h;fila final en dh, columna final en cl

 int 10h;ejecuta las interrupciones de video

 ret

limpiarP endp

Fin proc near ;Procedimiento finalizar el programa

 Puntero 15

 Mensaje txt1

 mov ah,04ch

 int 21h

 ret

Fin endp

CODE ENDS

END INICIO

MACROS-PROCEDIMIENTOS.ASM

;@Manuel Uc Nicoli

;Macros y procedimientos para el uso de las banderas.

;----- MACROS -----

Mensaje macro txt ;macro para mostrar textos desde una variable del programa.

```
    mov ah,09h
    lea dx, txt
    int 21h
endm
```

BanderaSuperior macro ;macro bandera superior.

```
    mov ax,0600h
    mov bh,17h;atributos de color fondo y texto
    mov CX,0000h;fila inicial en ch, columna inicial en cl
    mov dh, 4h
    mov dl, 13h
    int 10h;ejecuta las interrupciones de video
endm
```

BanderaInferior macro ;macro bandera inferior.

```
    mov ax,0600h
    mov bh,0E7h;atributos de color fondo y texto
    mov ch,5h;fila inicial en ch, columna inicial en cl
    mov cl,0h
```

```
    mov dh,9h;fila final en dh, columna final en cl
    mov dl,13h
    int 10h;ejecuta las interrupciones de video
endm
```

Puntero macro num1 ;macro para poner el puntero en una posicion hacia abajo.

```
    mov ah,02h
    mov dh,num1
    mov dl,2h
    mov bh,0h
    int 10h
endm
```

Puntero2 macro num1,num2 ;macro para poner el puntero en una posicion hacia abajo,derecha.

```
    mov ah,02h
    mov dh,num1
    mov dl,2h
    mov bh,num2
    int 10h
endm
```

;----- PROCEDIMIENTOS -----

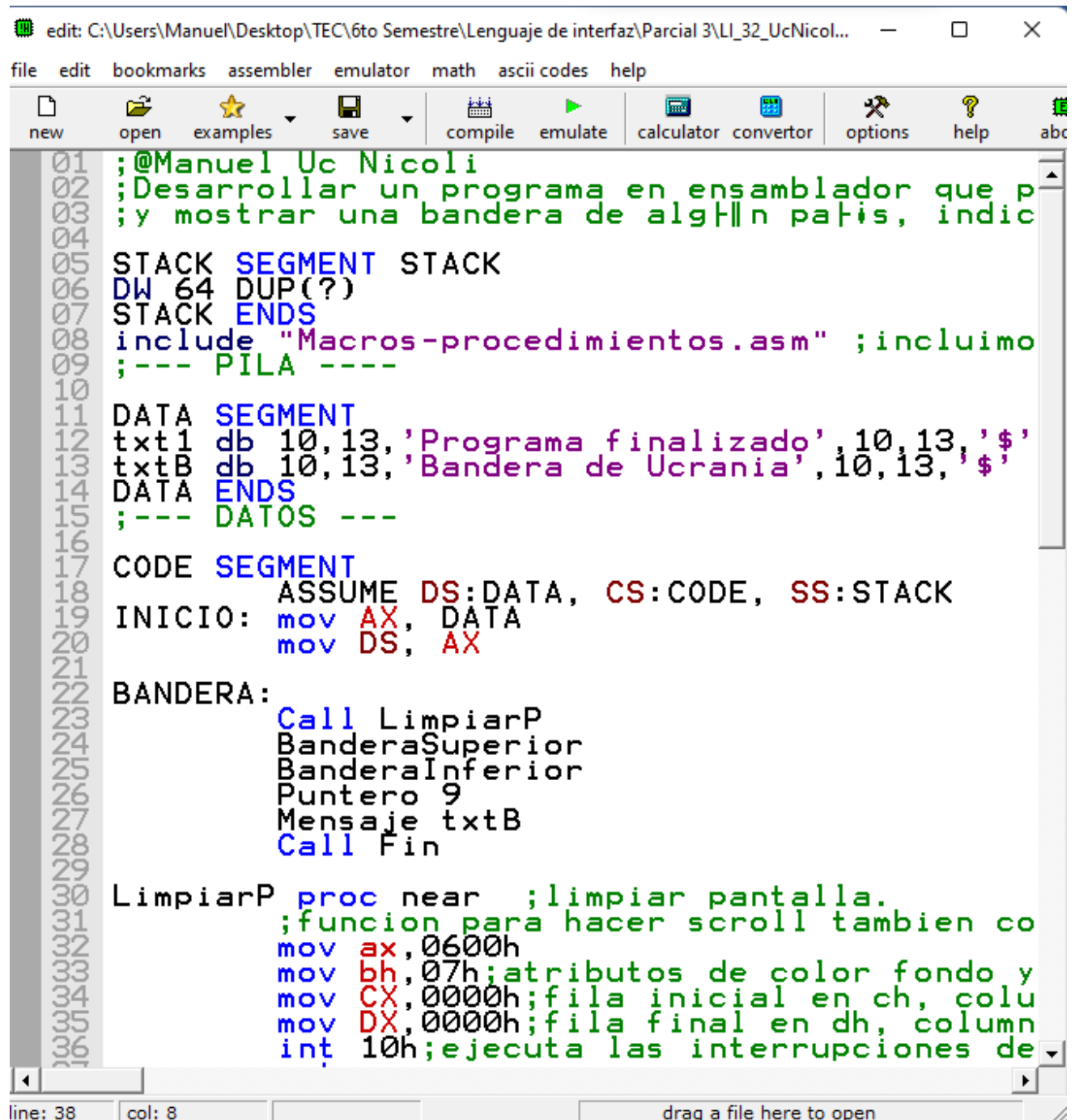
introducir proc far ;procedimeinto para pedir "x" tecla.

```
    mov ah,01h
```

```
int 21h          ;Coloca la tecla introducida en AL.  
mov dl, al       ;Colocamos en dl el valor de AL.  
ret  
introducir endp
```

Funcionalidad realizada en EMU8086 v4.08:

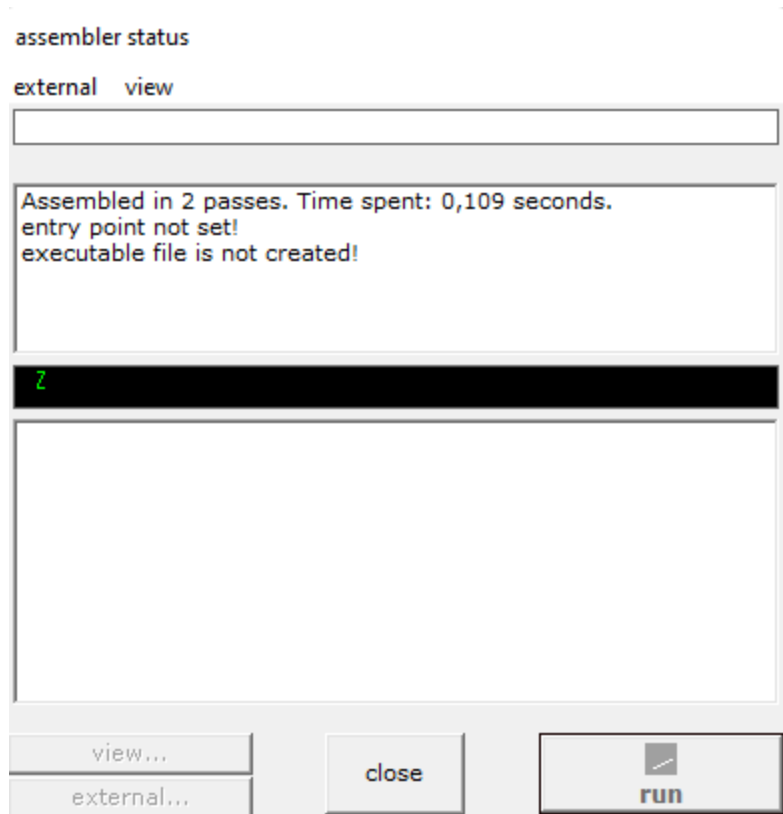
Carga del archivo dentro del programa:



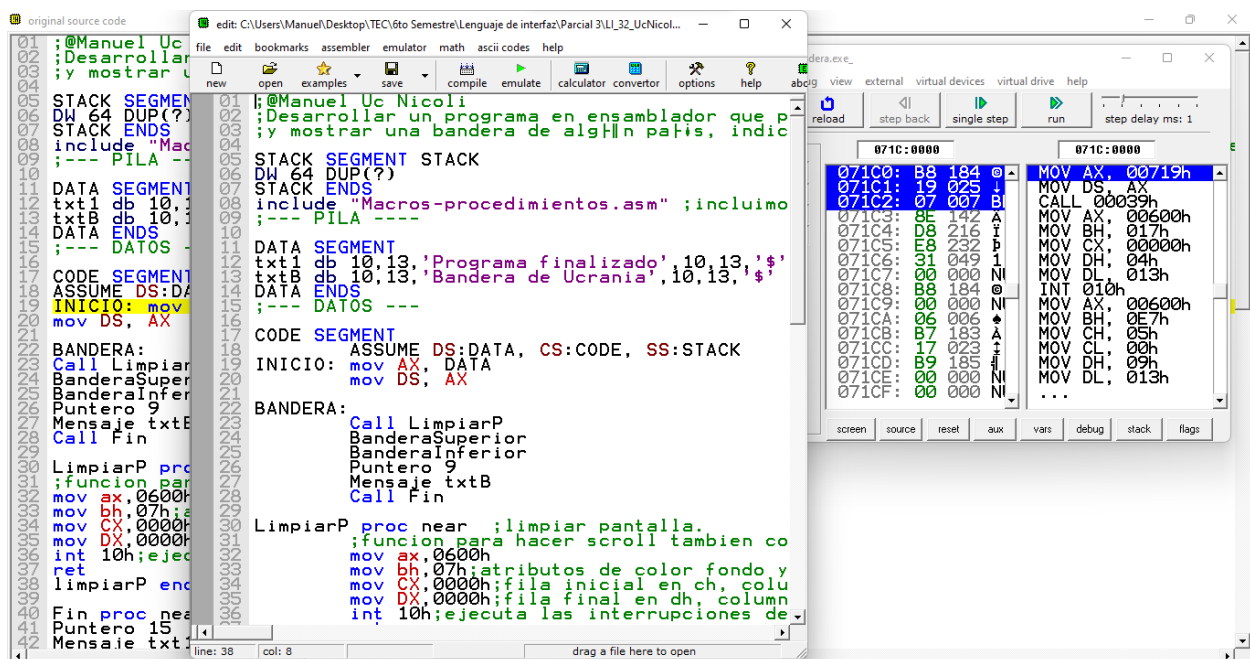
```
01 ;@Manuel Uc Nicoli
02 ;Desarrollar un programa en ensamblador que p
03 ;y mostrar una bandera de algñn país, indic
04
05 STACK SEGMENT STACK
06 DW 64 DUP(?)
07 STACK ENDS
08 include "Macros-procedimientos.asm" ;incluimo
09 ;--- PILA ---
10
11 DATA SEGMENT
12 txt1 db 10,13,'Programa finalizado',10,13,'$'
13 txtB db 10,13,'Bandera de Ucrania',10,13,'$'
14 DATA ENDS
15 ;--- DATOS ---
16
17 CODE SEGMENT
18 ASSUME DS:DATA, CS:CODE, SS:STACK
19 INICIO: mov AX, DATA
20         mov DS, AX
21
22 BANDERA:
23         Call LimpiarP
24         BanderaSuperior
25         BanderaInferior
26         Puntero 9
27         Mensaje txtB
28         Call Fin
29
30 LimpiarP proc near ;limpiar pantalla.
31             ;funcion para hacer scroll tambien co
32             mov ax,0600h
33             mov bh,07h;atributos de color fondo y
34             mov CX,0000h;fila inicial en ch, colu
35             mov DX,0000h;fila final en dh, column
36             int 10h;ejecuta las interrupciones de
37
```

line: 38 col: 8 drag a file here to open

Compilación del archivo para convertirlo en .exe (ejecutable) sin ningún problema u error.



Pestañas abiertas durante la ejecución del programa (vista):



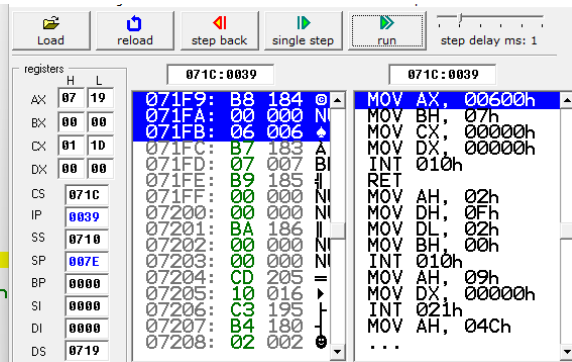
Instrucciones realizadas paso a paso.

Se realiza de manera satisfactoria la limpieza de la pantalla por medio de un procedimiento que ejecuta una interrupción 10H.

```
CODE SEGMENT
ASSUME DS:DATA, CS:CODE, SS:STACK
INICIO: mov AX, DATA
mov DS, AX

BANDERA:
Call LimpiarP
BanderaSuperior
BanderaInferior
Puntero 9
Mensaje txtB
Call Fin

LimpiarP proc near ;limpiar pantalla,
;funcion para hacer scroll tambien con 7h
mov ax, 0600h
mov bh, 07h;atributos de color fondo y texto
mov cx, 0000h;fila inicial en ch, columna inicial en cl
mov dx, 0000h;fila final en dh, columna final en cl
int 10h;ejecuta las interrupciones de video
ret
LimpiarP endp
```



Posteriormente llamamos una macro donde se encuentra la parte superior de nuestra bandera y luego llamamos la parte inferior del color de nuestra bandera.

```
24 BanderaSuperior
25 BanderaInferior
```

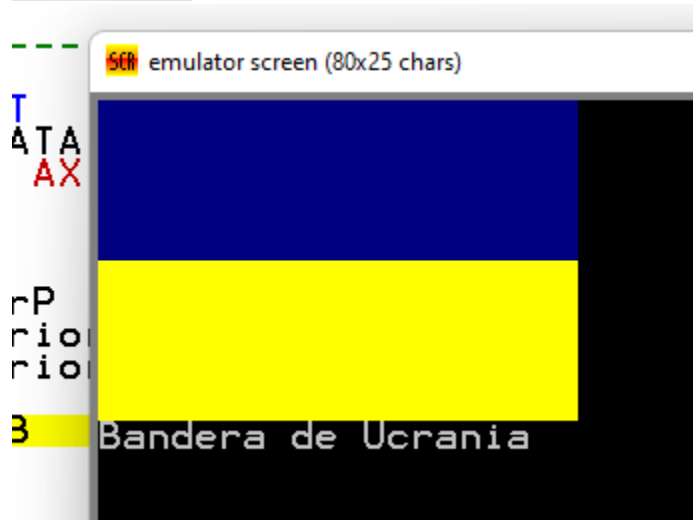
Seguidamente pasamos a la siguiente macro en este caso, utiliza la misma interrupción 10H de video, en este caso colocaremos de manera satisfactoria el puntero donde iniciara el mensaje en este caso la macro pide un numero el cual lo pasa al otro .asm por lo que este proceso no se puede ver a simple vista.

```
26 Puntero 9
```

```
30 Puntero macro num1
31     mov ah, 02h
32     mov dh, num1
33     mov dl, 2h
34     mov bh, 0h
35     int 10h
36 endm
```

Luego tenemos el macro para imprimir texto donde este igual necesita de una variable, en este caso en formato de cadena, por lo que seguidamente nos desplegara la ventana ya con la ventana creada, para posteriormente finalizar el programa.

```
27 Mensaje txtB
5 Mensaje macro txt
6     mov ah,09h
7     lea dx, txt
8     int 21h
9 endm
```



Por ultimo llamamos el procedimiento el cual finaliza el programa y dentro de este mismo utiliza de nueva cuenta la macro donde colocamos el puntero y la macro que imprimirá el mensaje donde ha finalizado el programa.

```
40 Fin proc near
41 Puntero 15
42 Mensaje txt1
43 mov ah,04ch
44 int 21h
45 ret
46 Fin endp
```

Finalización del programa.

