

Actividad de
Luis Enrique Hernandez Marin



Lenguajes de Interfaz

T1P1 Hola Mundo

Tema 1

Profesor Luis Humberto Sánchez Medel

Semestre 6

Fecha: 29 / 03 / 2025



Practica 1.1: HOLA MUNDO

Objetivo: El alumno desarrollara un programa en ensamblador

MATERIAL Y EQUIPO

- Computadora
- Software EMU8086

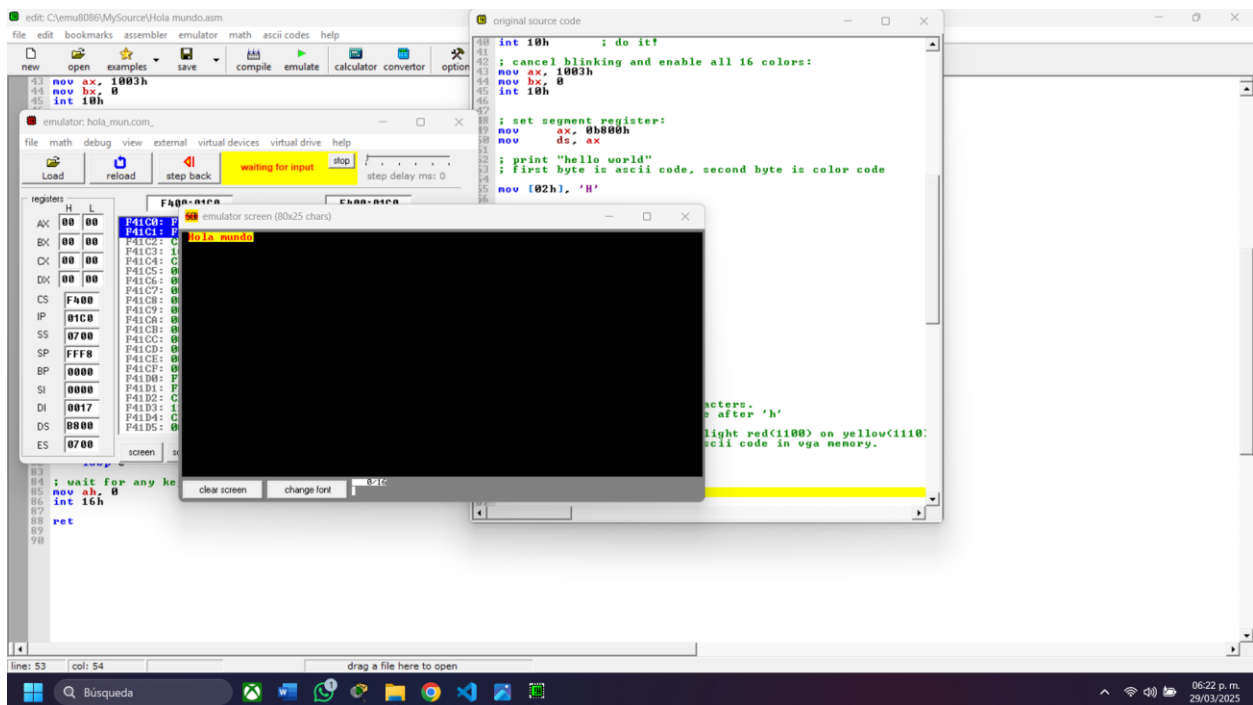
PROCEDIMIENTO DE LA PRACTICA

- a) Escriba el programa anexo a este manual
- b) Ensamble el código y corra el programa

PRUEBAS Y EVIDENCIAS

Captura de pantalla del programa funcionando

Hola mundo



Código (reducir tamaño, doble columna y quitar espacio)

Escribir el código en Emu80086 otorgado por el profesor el cual consiste en imprimir un hola mundo el cual quedo de la siguiente manera:

```
edit: C:\emu8086\MySource\Hola mundo.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

;PROGRAMA: HOLA MUNDO
name "Hola mundo"

; this example prints out "hello world!"
; by writing directly to video memory.
; in vga memory: first byte is ascii character,
; byte that follows is character attribute.
; if you change the second byte, you can
; change the color of
; the character even after it is printed.
; character attribute is 8 bit value.
; high 4 bits set background color and low 4 bits set foreground color.

; hex bin color
; 0 0000 black
; 1 0001 blue
; 2 0010 green
; 3 0011 cyan
; 4 0100 red
; 5 0101 magenta
; 6 0110 brown
; 7 0111 light gray
; 8 1000 dark gray
; 9 1001 light blue
; a 1010 light green
; b 1011 light cyan
; c 1100 light red
; d 1101 light magenta
; e 1110 yellow
; f 1111 white

org 100h
; set video mode
mov ax, 3 ; text mode 80x25, 16 colors, 8
; pager (ah=0, al=3)
int 10h ; do it!

; cancel blinking and enable all 16 colors:
mov ax, 1003h
mov bx, 0
int 10h

; set segment register:
mov ax, 0b800h
mov ds, ax

; print "hello world"
; first byte is ascii code, second byte is color code
mov [02h], 'H'
mov [04h], 'o'
```

```
edit: C:\emu8086\MySource\Hola mundo.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

mov ax, 1003h
mov bx, 0
int 10h

; set segment register:
mov ax, 0b800h
mov ds, ax

; print "hello world"
; first byte is ascii code, second byte is color code
mov [02h], 'H'
mov [04h], 'o'
mov [06h], 'l'
mov [08h], 'o'
mov [0ah], ' '
mov [0ch], 'n'
mov [0eh], 'u'
mov [08h], 'n'
mov [10h], 'd'
mov [12h], 'd'
mov [14h], 'o'

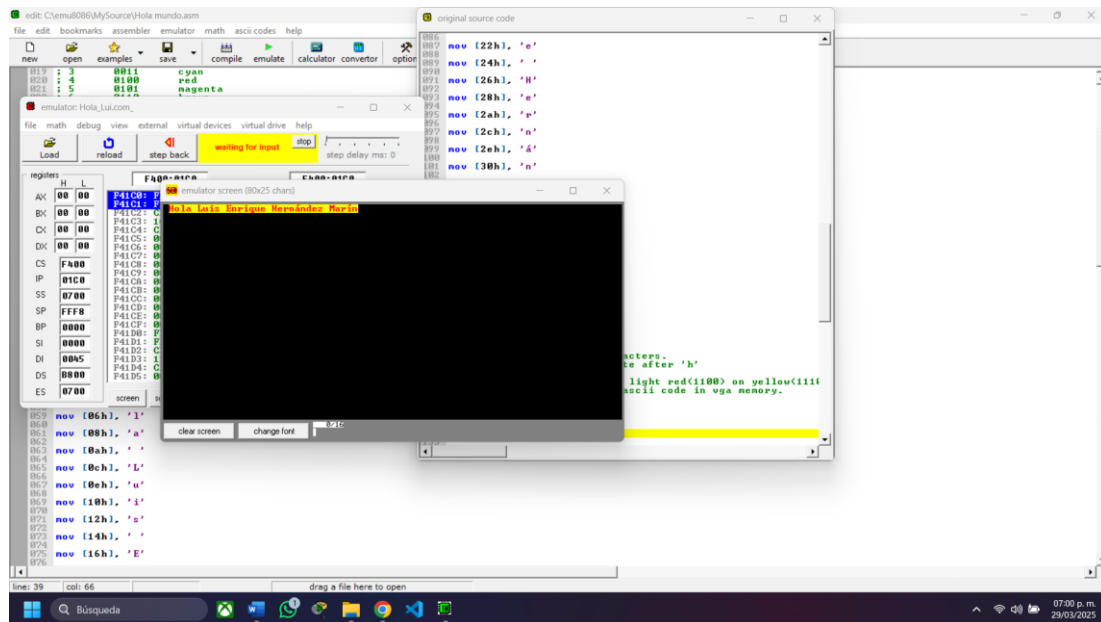
; color all characters:
mov cx, 10 ; number of characters.
mov di, 03h ; start from byte after 'h'
c: mov [di], 1101100b ; light red(1100) on yellow(1110)
add di, 2 ; skip over next ascii code in vga memory.
loop c

; wait for any key press:
mov ah, 0
int 16h
ret
```

Mostrar en ves de “Hola mundo” debe mostrar el mensaje “Hola TUNOMBRECOMPLETO” en mi caso seria

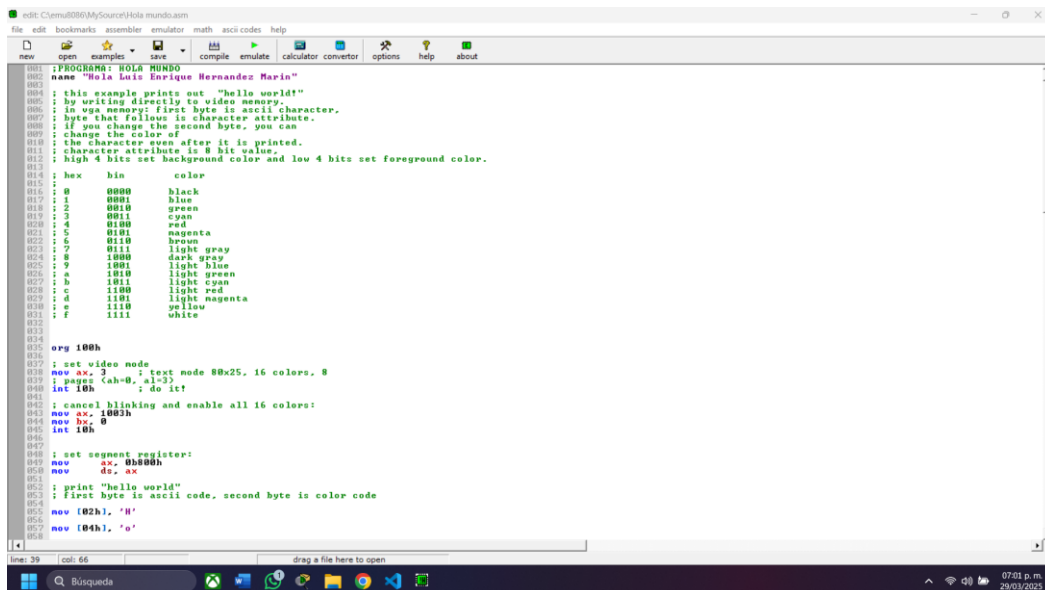
“Hola Luis Enrique Hernández Marín”

Ejecución del programa



Código (reducir tamaño, doble columna y quitar espacio)

Modificar el código en Emu8086 otorgado por el profesor el cual consiste en imprimir un “hola mundo” a “Hola Luis Enrique Hernández Marín” el cual quedo de la siguiente manera:



```
edit: C:\emu8086\MySource\Hola mundo.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
0052 ; print "hello world"
0053 ; first byte is ascii code, second byte is color code
0054
0055 mov [02h], 'H'
0056
0057 mov [04h], 'o'
0058
0059 mov [06h], 'l'
0060
0061 mov [08h], 'a'
0062
0063 mov [0ah], ' '
0064
0065 mov [0ch], 'l'
0066
0067 mov [0eh], 'u'
0068
0069 mov [10h], 'i'
0070
0071 mov [12h], 's'
0072
0073 mov [14h], ' '
0074
0075 mov [16h], 'E'
0076
0077 mov [18h], 'n'
0078
0079 mov [1ah], 'r'
0080
0081 mov [1ch], 'i'
0082
0083 mov [1eh], 'q'
0084
0085 mov [20h], 'u'
0086
0087 mov [22h], 'e'
0088
0089 mov [24h], ' '
0090
0091 mov [26h], 'H'
0092
0093 mov [28h], 'e'
0094
0095 mov [2ah], 'r'
0096
0097 mov [2ch], 'n'
0098
0099 mov [2eh], 'á'
0100
0101 mov [30h], 'n'
0102
0103 mov [32h], 'd'
0104
0105 mov [34h], 'e'
0106
0107 mov [36h], 'z'
0108
0109 mov [38h], ' '
line: 39 col: 66 drag a file here to open
Búsqueda 07:02 p. m. 29/03/2025
```

```
edit: C:\emu8086\MySource\Hola mundo.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
1105
1107 mov [36h], 'z'
1108
1109 mov [38h], ' '
1110
1111 mov [3ah], 'M'
1112
1113 mov [3ch], 'a'
1114
1115 mov [3eh], 'r'
1116
1117 mov [40h], 'i'
1118
1119 mov [42h], 'n'
1120
1121
1122 ; color all characters:
1123 mov cx, 33 ; number of characters.
1124 mov di, 03h ; start from byte after 'h'
1125
1126 c: mov [di], 11101100b ; light red(1100) on yellow(1110)
1127 add di, 2 ; skip over next ascii code in vga memory.
1128 loop c
1129
1130 ; wait for any key press:
1131 mov ah, 0
1132 int 16h
1133
1134 ret
1135
1136
line: 39 col: 66 drag a file here to open
Búsqueda 07:03 p. m. 29/03/2025
```

```

;PROGRAMA: HOLA MUNDO

name "Hola Luis Enrique Hernandez Marin"

; this example prints out "hello world!"

; by writing directly to video memory.

; in vga memory: first byte is ascii character,
; byte that follows is character attribute.

; if you change the second byte, you can
; change the color of
; the character even after it is printed.

; character attribute is 8 bit value,
; high 4 bits set background color and low 4 bits
set foreground color.

; hex  bin    color
;
; 0  0000  black
; 1  0001  blue
; 2  0010  green
; 3  0011  cyan
; 4  0100  red
; 5  0101  magenta
; 6  0110  brown
; 7  0111  light gray
; 8  1000  dark gray
; 9  1001  light blue
; a  1010  light green
; b  1011  light cyan
; c  1100  light red
; d  1101  light magenta
; e  1110  yellow

```

```

; f  1111  white

org 100h

; set video mode

mov ax, 3 ; text mode 80x25, 16 colors, 8

; pages (ah=0, al=3)

int 10h ; do it!

; cancel blinking and enable all 16 colors:

mov ax, 1003h

mov bx, 0

int 10h

; set segment register:

mov ax, 0b800h

mov ds, ax

; print "Hola Luis Enrique Hernandez Marin"

; first byte is ascii code, second byte is color
code

mov [02h], 'H'

mov [04h], 'o'

mov [06h], 'l'

mov [08h], 'a'

mov [0ah], ' '

mov [0ch], 'L'

mov [0eh], 'u'

mov [10h], 'i'

mov [12h], 's'

mov [14h], ' '

mov [16h], 'E'

mov [18h], 'n'

mov [1ah], 'r'

```

| | |
|----------------|--|
| mov [1ch], 'i' | mov [3ch], 'a' |
| mov [1eh], 'q' | mov [3eh], 'r' |
| mov [20h], 'u' | mov [40h], 'i' |
| mov [22h], 'e' | mov [42h], 'n' |
| mov [24h], ' ' | ; color all characters: |
| mov [26h], 'H' | mov cx,33 ; number of characters. |
| mov [28h], 'e' | mov di, 03h ; start from byte after 'h' |
| mov [2ah], 'r' | c: mov [di], 11101100b ; light red(1100) on |
| mov [2ch], 'n' | yellow(1110) |
| mov [2eh], ' ' | add di, 2 ; skip over next ascii code in vga |
| mov [30h], 'n' | memory. |
| mov [32h], 'd' | loop c |
| mov [34h], 'e' | ; wait for any key press: |
| mov [36h], 'z' | mov ah, 0 |
| mov [38h], ' ' | int 16h |
| mov [3ah], 'M' | ret |

RESULTADOS & CONCLUSIONES

Escriba a que conclusión llegaron las pruebas realizadas

Con las pruebas realizadas he llegado a la conclusión que para tan siquiera mostrar solamente un mensaje se debe de programar bastante código, en donde se deben manejar los registros y algunos comandos como MOV y también que para manejar la paleta de colores que había en el código es necesario serie de 0 y 1, en donde se puede establecer el fondo y el color del texto.

Además, que el manejo de MOV para asignar las letras que en este caso fue el de hola y mi nombre completo seguía un patrón que debía cumplir por lo se lee en los comentarios el primer byte era del código ASCII y el segundo byte el código de color.

Puedo llegar a mencionar que es interesante programar en lenguaje ensamblador que un lenguaje de bajo nivel, para observar como se maneja el código de alto nivel para que llegue a la computadora